Red Hat Linux 9

Guide de référence de Red Hat Linux



Red Hat Linux 9: Guide de référence de Red Hat Linux



Red Hat, Inc.

1801 Varsity Drive
Raleigh NC 27606-2072 USA
Phone: +1 919 754 3700
Phone: 888 733 4281
Fax: +1 919 754 3701
PO Box 13588
Research Triangle Park NC 27709 USA

rhl-rg(FR)-9-Print-RHI (2003-02-13T19:20)

Le code GPG de la clé security@redhat.com key est:

CA 20 86 86 2B D6 9D FC 65 F6 EC C4 21 91 80 CD DB 42 A6 0E

Table des matières



Introduction

Bienvenue dans le *Guide de référence de Red Hat Linux*.

Le *Guide de référence de Red Hat Linux* contient des informations utiles sur le système Red Hat Linux. Depuis les concepts fondamentaux tels que la structure des systèmes de fichiers de Red Hat Linux, jusqu'à certains points plus délicats concernant la sécurité du système et le contrôle de l'authentification, nous espérons que ce guide sera pour vous une précieuse ressource.

Ce guide vous convient tout particulièrement si vous souhaitez en savoir plus sur la manière dont fonctionne votre système Red Hat Linux. Il examine en effet, les sujets suivants:

- La structure du système de fichiers;
- Le processus de démarrage;
- Le système X Window;
- Les outils de sécurité;
- Les services de réseau.

1. Modifications apportées à ce manuel

La structure de ce manuel a été réorganisée dans un souci de clarté. Le manuel a également été mis à jour de manière à inclure les nouvelles fonctionnalités de Red Hat Linux 9. Ci-après figure une liste des modifications apportées:

Mise à jour du chapitre Système X Window

Le chapitre *Système X Window* a été complètement révisé et réorganisé dans un souci de clarté. En outre, de nouvelles instructions concernant la configuration des polices de caractères ont été ajoutées.

Nouveau chapitre `sysconfig`

La section `sysconfig` du chapitre *Processus de démarrage, Init, arrêt* a non seulement été étoffée mais convertie en un chapitre à part entière.

Mise à jour du chapitre Enveloppeurs TCP et `xinetd`

Le chapitre *Envelopeurs TCP et* `xinetd` révisé a été restructuré dans un souci de clarté.

Mise à jour du chapitre Utilisateurs et groupes

Le chapitre *Utilisateurs et groupes* a été mis à jour et restructuré pour permettre une meilleure compréhension.

Mise à jour du chapitre Interfaces réseau

Le chapitre *Interfaces réseau* a été mis à jour et réorganisé.

Mise à jour du chapitre Serveur HTTP Apache

Le guide de migration de la version 1.3 vers la version 2.0 de Serveur HTTP Apache a été révisé. La liste des options de configuration de serveur a également été mise à jour et réorganisée. Nous remercions tout spécialement **Gary Benson** et **Joe Orton** pour leur contribution à ce guide traitant de la migration du Serveur HTTP Apache.

Avant d'entamer la lecture de ce guide, vous devriez connaître les aspects concernant l'installation qui sont reportés dans le *Guide d'installation de Red Hat Linux*, les concepts de base de Linux qui sont

contenus dans le *Guide de démarrage de Red Hat Linux* et et les instructions générales de personnalisation qui sont décrites dans le *Guide de personnalisation de Red Hat Linux*. Le *Guide de référence de Red Hat Linux* contient des informations plus complexes pour les utilisateurs expérimentés.

Les versions HTML et PDF de tous les manuels de Red Hat Linux sont disponibles en ligne à l'adresse suivante: http://www.redhat.com/docs



Remarque

Bien que le présent manuel contienne les informations les plus actuelles possibles, il est recommandé de lire les *Notes de mises à jour* de Red Hat Linux au cas où de nouvelles informations auraient été ajoutées après l'impression de cette documentation. Les *Notes de mise à jour* se trouvent sur le CD-ROM 1 de Red Hat Linux et en ligne à l'adresse suivante:

`http://www.redhat.com/docs/manuals/linux`

2. Documentation appropriée à vos besoins

Il est essentiel que vous disposiez d'une documentation appropriée à votre niveau de maîtrise de Linux. En effet, dans le cas contraire, vous vous sentirez peut-être dépassé ou vous ne pourrez pas trouver les informations nécessaires pour répondre à vos questions. Le *Guide de référence de Red Hat Linux* traite des aspects et des options les plus techniques de votre système Red Hat Linux. Cette section vous aidera à décider si ce manuel répondra à vos questions ou si vous devez consulter d'autres guides Red Hat Linux, y compris les ressources disponibles en ligne.

Passons en revue les trois catégories d'utilisateurs de Red Hat Linux et déterminons la documentation dont ils ont besoin. Commençons par déterminer votre niveau d'expérience. Ci-dessous figurent trois catégories de base:

Débutant

Personne n'ayant jamais, ou presque jamais, utilisé un système d'exploitation Linux (ou analogue). Personne pouvant éventuellement avoir déjà utilisé d'autres systèmes d'exploitation (tels que Windows). Est-ce votre cas? Si oui, reportez-vous à la la Section 2.1.

Moyennement expérimenté

Personne ayant déjà installé et utilisé Linux (mais pas Red Hat Linux) avec succès auparavant. Ou alors, personne disposant d'une expérience équivalente avec d'autres systèmes d'exploitation de type Linux. Est-ce votre cas? Si oui, reportez-vous à la documentation de la la Section 2.2.

Chevronné

Personne ayant déjà installé et utilisé Red Hat Linux avec succès précédemment. Est-ce votre cas? Si oui, reportez-vous à la la Section 2.3.

2.1. Documentation pour les débutants

Pour un nouveau-venu au monde Linux, la quantité d'informations disponibles sur des sujets de base tels que l'impression, le démarrage du système ou le partitionnement du disque dur est impressionnante. Ces informations permettent d'acquérir de solides bases sur le fonctionnement de Linux, avant d'approfondir des sujets plus avancés.

Commencez par vous procurer la documentation adéquate. On ne soulignera jamais assez l'importance de cette étape. En effet, sans documentation vous ne pourrez qu'être frustré en raison de votre incapacité à faire fonctionner le système Red Hat Linux comme vous le souhaiteriez.

Ci-après figure une liste du type de documentation Linux que vous devriez avoir sous la main:

- *Bref historique de Linux* — De nombreux aspects de Linux sont le fruit d'une évolution. Il existe également une culture Linux qui, une fois encore, puise largement dans son passé. Quelques connaissances concernant l'histoire de Linux vous seront utiles, en particulier pour apprendre à résoudre beaucoup de problèmes potentiels avant leur apparition.
- *S'il n'est pas indispensable de maîtriser tous les aspects du noyau Linux,* il est utile de savoir de quoi Linux est fait. Ce point est particulièrement important si vous avez déjà travaillé avec d'autres systèmes d'exploitation; certaines de vos certitudes quant au fonctionnement des ordinateurs peuvent ne pas être transposables à Linux.
- *Aperçu des commandes (avec des exemples)* — Ce document est probablement l'élément le plus important de la documentation de Linux. La philosophie de conception sous-jacente à Linux est qu'il est préférable d'utiliser de nombreuses petites commandes interconnectées de différentes manières plutôt que d'avoir un grand nombre de commandes volumineuses (et complexes) qui font tout le travail. Si vous ne disposez pas d'exemples illustrant cette approche de Linux, vous risquez d'être effrayé rien que par le nombre de commandes disponibles sur votre système Red Hat Linux.

  Souvenez-vous que vous ne devez pas connaître toutes les commandes Linux existantes. Différentes techniques permettent de trouver la commande requise pour l'accomplissement d'une tâche. Vous devez simplement comprendre le fonctionnement de Linux de façon générale, ce que vous devez accomplir et comment accéder à l'outil qui vous fournira les instructions exactes permettant l'exécution de la commande.

Le *Guide d'installation de Red Hat Linux* est une excellente référence qui vous assistera dans l'installation et la configuration initiale de Red Hat Linux. Le *Guide de démarrage de Red Hat Linux* couvre les commandes de base du système, l'environnement de bureau graphique et bien d'autres concepts fondamentaux. Nous vous conseillons de commencer par ces deux livres afin d'acquérir vos connaissances de base sur Red Hat Linux. Il ne vous faudra pas beaucoup de temps avant que des concepts plus compliqués ne deviennent très clairs, car vous aurez compris les idées principales de Linux.

Outre les manuels Red Hat Linux, bien d'autres sources de documentation sont disponibles à un prix réduit ou gratuitement. Parmi celles-ci figurent entre autres:

2.1.1. Introduction aux sites Web de Linux

- http://www.redhat.com — Sur le site Web de Red Hat vous trouverez des liens qui vous permettront de consulter le Projet de documentation Linux (LDP, Linux Documentation Project), les versions en ligne des manuels Red Hat Linux, le Forum Aux Questions (FAQ), une base de données qui vous assiste dans la recherche d'un Groupe d'utilisateurs Linux près de chez vous, les informations techniques contenues dans le Red Hat Support Knowledge Base, etc.
- http://www.linuxheadquarters.com — Le site Web du siège social de Linux contient de nombreux guides examinant différents outils de Linux.

2.1.2. Introduction aux groupes de discussion Linux

Vous pouvez participer aux groupes de discussion en suivant les interventions d'autres personnes, en posant des questions ou en essayant de répondre aux questions posées. Les utilisateurs expérimentés de Linux sont passés maîtres dans l'art d'aider les débutants à comprendre Linux — en particulier si les questions sont bien formulées et adressées au forum approprié. Si vous n'avez pas accès à une application qui permet d'entrer dans ces groupes, vous pouvez accéder à ces informations sur le Web à

l'adresse http://groups.google.com/. Il existe des dizaines de groupes de discussion concernant Linux; parmi ceux-ci figurent:

- linux.help — Un excellent site où vous obtiendrez de l'aide de la part d'autres utilisateurs Linux.
- linux.redhat — Ce groupe de discussion aborde des thèmes spécifiques à Red Hat Linux.
- linux.redhat.install — Posez vos questions concernant l'installation ou voyez comment d'autres personnes résolvent des problèmes similaires aux vôtres.
- linux.redhat.misc — Pour des questions ou des demandes d'aide particulières.
- linux.redhat.rpm — Une bonne adresse si vous n'arrivez pas à atteindre des objectifs particuliers avec **RPM**.

2.1.3. Livres sur Linux pour les utilisateurs débutants

- *Red Hat Linux for Dummies, 2ème édition* de Jon "maddog" Hall, édité par IDG
- *Special Edition Using Red Hat Linux* de Alan Simpson, John Ray et Neal Jamison, édité par Que
- *Running Linux* de Matt Welsh et Lar Kaufman, édité par O'Reilly & Associates
- *Red Hat Linux 8 Unleashed* de Bill Ball et Hoyle Duff; Pearson Education

Les livres ci-dessus sont d'excellentes sources d'informations sur le fonctionnement de base du système Red Hat Linux. Pour des informations plus approfondies, reportez-vous aux livres mentionnés dans les différents chapitres de ce manuel, en particulier dans la section *Ressources supplémentaires*.

2.2. Documentation pour les utilisateurs expérimentés

Si vous avez utilisé d'autres distributions Linux, vous connaissez probablement déjà les commandes les plus utilisées. Vous avez peut être installé votre système Linux et téléchargé des logiciels que vous avez trouvés sur Internet. Une fois Linux installé, les procédures de configuration peuvent toutefois poser problème.

Le *Guide de personnalisation de Red Hat Linux* est conçu pour expliquer la ou les configuration(s) du système Red Hat Linux afin de pouvoir choisir celle répondant le mieux à vos objectifs. Ce guide vous permettra d'acquérir des connaissances sur des options de configuration spécifiques et vous expliquera comment les appliquer.

Lorsque vous installez des logiciels qui ne figurent pas dans le *Guide de personnalisation de Red Hat Linux*, il est souvent utile de voir ce que d'autres personnes ont fait dans des circonstances similaires. Les documents HOWTO du Projet de documentation Linux, disponibles à l'adresse suivante: http://www.redhat.com/mirrors/LDP/HOWTO/HOWTO-INDEX/howtos.html, traitent des aspects particuliers de Linux, des modifications ésotériques du noyau de bas niveau à l'utilisation de Linux pour des stations de radio-amateurs.

2.3. Documentation pour les utilisateurs chevronnés

Si vous utilisez Red Hat Linux depuis longtemps, vous savez probablement que le meilleur moyen de comprendre un programme est de lire son code source et/ou ses fichiers de configuration. L'un des plus principaux avantages de Red Hat Linux est que le code source est toujours disponible.

Évidemment, comme nous ne sommes pas tous des programmateurs, le code source ne sera pas forcément d'une grande aide. Toutefois, si vous avez les connaissances et les aptitudes nécessaires pour le comprendre, le code source peut répondre à toutes vos interrogations.

3. Conventions de documentation

En lisant ce manuel vous verrez que certains mots sont représentés avec des polices différentes au niveau du type, de la taille et de l'utilisation de caractères gras. Cette présentation est systématique; différents mots sont représentés dans le même style pour indiquer leur appartenance à une certaine catégorie. Parmi les types de mots représentés de cette façon figurent:

`commande`

Les commandes de Linux (et les commandes d'autres systèmes d'exploitation, lorsqu'elles sont utilisées) sont représentées de cette façon. Ce style vous indique que vous pouvez taper le mot ou l'expression sur la ligne de commande et appuyer sur [Entrée] pour invoquer une commande. Une commande contient parfois des mots qui, tous seuls, seraient représentés différemment (comme les noms de fichiers). Dans ces cas là, ils sont considérés comme une partie de la commande; toute la phrase sera donc affichée comme une commande. Par exemple:

Utilisez la commande `cat fichier_test` pour afficher le contenu d'un fichier, nommé `fichier_test`, dans le répertoire de travail courant.

`nom de fichier`

Les noms de fichiers, de répertoires, les chemins d'accès et les noms de paquetages RPM sont représentés de cette façon. Ce style devrait indiquer qu'un fichier ou un répertoire de ce nom existe dans votre système Red Hat Linux. Exemples:

Le fichier `.bashrc` dans votre répertoire personnel contient des définitions et alias de shell bash pour votre utilisation personnelle.

Le fichier `/etc/fstab` contient les informations concernant les différents périphériques et systèmes de fichiers du système.

Installez le RPM `webalizer` si vous voulez utiliser un programme d'analyse de fichier journal de serveur Web.

application

Ce style indique que le programme est une application d'utilisateur final (au contraire de logiciels de système). Par exemple:

Utilisez **Mozilla** pour parcourir le Web.

[touche]

Une touche du clavier est représentée de cette façon. Par exemple:

Pour utiliser l'achèvement [Tab], tapez un caractère, puis appuyez sur la touche [Tab]. Votre terminal affichera la liste des fichiers du répertoire qui commencent avec cette lettre.

[touche]-[combinaison]

Une combinaison de touches est représentée de cette façon. Par exemple:

La combinaison [Ctrl]-[Alt]-[Effacement arrière] vous déconnecte de votre session graphique et revient sur l'écran de connexion graphique ou la console.

texte trouvé sur une interface GUI

Un titre, un mot ou une phrase trouvé sur l'écran ou la fenêtre d'une interface GUI est représenté de cette façon. Lorsque vous voyez du texte dans ce style, il est utilisé pour identifier un écran GUI ou un élément sur un écran GUI particulier (comme du texte associé avec une case à cocher ou un champ). Exemple:

Cochez la case **Nécessite un mot de passe** si vous voulez que votre écran de veille demande un mot de passe avant de s'arrêter.

premier niveau d'un menu sur un écran ou une fenêtre GUI

Ce style vous indique que le mot représente le premier élément d'un menu déroulant. Cliquez sur le mot de l'écran GUI pour afficher le reste du menu. Par exemple:

Sous **Fichier** d'un terminal GNOME, vous trouverez l'option **Nouvel onglet** vous permettant d'ouvrir plusieurs invites du shell dans la même fenêtre.

Si vous devez entrer une séquence de commandes depuis un menu GUI, elles apparaîtront de la façon suivante:

Cliquez sur **Menu principal** (sur le tableau de bord) => **Programmation** => **Emacs** pour lancer l'éditeur de texte **Emacs**.

bouton sur un écran ou une fenêtre GUI

Ce style indique que le texte se trouve sur un bouton à cliquer sur un écran GUI. Par exemple:

Cliquez sur le bouton **Retour** pour revenir à la dernière page Web que vous avez affichée.

`sortie d'ordinateur`

Du texte dans ce style vous indique qu'il est affiché par l'ordinateur en ligne de commande. Vous verrez affiché de cette manière les réponses aux commandes que vous avez tapées, des messages d'erreur et des invites interactives pour vos saisies durant des scripts ou des programmes. Par exemple:

Utilisez la commande `ls` pour afficher le contenu d'un répertoire:

```
$ls
Desktop             about.html     logs        paulwesterberg.png
Mail                backupfiles    mail        reports
```

La sortie produite en réponse à cette commande (dans ce cas, le contenu du répertoire) est affichée de cette façon.

`invite`

L'invite est la façon qu'a l'ordinateur de vous indiquer qu'il est prêt à recevoir votre saisie. Elle est représentée de cette façon. Exemples:

`$`

`#`

`[stephen@maturin stephen]$`

`leopard login:`

`saisie de l'utilisateur`

Le texte que l'utilisateur doit entrer, que ce soit en ligne de commande ou dans une zone de texte sur un écran GUI, est affiché de cette façon. Dans l'exemple suivant, **`text`** est affiché de cette façon:

Pour démarrer votre système dans le programme d'installation en mode texte, il vous faudra entrer la commande **`text`** à l'invite `boot:`.

De plus, nous utilisons différentes stratégies pour attirer votre attention sur certaines informations. Suivant l'importance de l'information pour votre système, ces éléments seront présentés sous forme de remarques, astuces, avertissements, messages importants ou attention. Par exemple:



Remarque

N'oubliez pas que Linux différencie les majuscules et les minuscules. Autrement dit, rose n'est ni ROSE ni rOsE.



Astuce

Le répertoire `/usr/share/doc` contient de la documentation supplémentaire pour les paquetages installés sur votre système.



Important

Si vous modifiez le fichier de configuration DHCP, les changements ne prendront pas effet tant que vous n'aurez pas redémarrer le démon DHCP.



Attention

N'effectuez pas de tâches quotidiennes en tant que root — utilisez un compte utilisateur normal à moins que vous n'ayez besoin d'utiliser le compte super-utilisateur pour des tâches d'administration système.



Avertissement

Si vous choisissez de ne pas partitionner manuellement, une installation serveur effacera toutes les partitions existantes sur tous les disques durs installés. N'utilisez cette classe d'installation que si vous êtes certain de ne pas avoir de données à sauvegarder.

4. Utilisation de la souris

Red Hat Linux utilise habituellement une souris à trois boutons. Si vous avez une souris à deux boutons, vous devriez avoir sélectionné l'émulation durant le processus d'installation. Si vous utilisez l'émulation de souris à trois boutons, cliquer simultanément sur les deux boutons revient à cliquer sur le bouton central (que vous n'avez pas).

Si le système vous demande de cliquer à un endroit, il est entendu qu'il s'agit du bouton gauche. Si vous devez utiliser le bouton central ou celui de droite, cela vous sera précisé. (Si vous avez configuré votre souris pour un gaucher, inversez ces instructions.)

L'expression "glisser et poser" (ou 'déplacement par glissement') vous est peut-être familière. Si vous devez glisser et poser un élément sur votre bureau d'interface graphique, cliquez sur cet élément et maintenez le bouton de la souris appuyé. Glissez ensuite l'élément, tout en maintenant la touche appuyée, vers son nouvel emplacement. Relâchez ensuite le bouton et posez l'élément.

5. Fonction 'Copier-coller' avec X

Il est facile de copier et coller du texte à l'aide de votre souris et du système X Window. Pour copier du texte, il vous suffit de cliquer et glisser votre souris sur le texte pour le mettre en surbrillance. Pour coller du texte, il suffit de cliquer avec le bouton central de la souris à l'endroit où vous voulez le placer.

6. Prochainement

Le *Guide de référence de Red Hat Linux* fait partie de l'engagement pris par Red Hat de fournir une assistance utile et ponctuelle aux utilisateurs Red Hat Linux. Les prochaines éditions contiendront de plus amples informations sur les changements de la structure et de l'organisation du système, de nouveaux outils de sécurité plus performants et d'autres ressources qui vous aideront à accroître la puissance de votre système Red Hat Linux — ainsi que vos capacités à l'exploiter au maximum de ses possibilités.

Pour nous permettre de remplir notre engagement, votre contribution est importante.

6.1. Vos commentaires sont importants

Si vous trouvez une erreur, faute de frappe dans le *Guide de référence de Red Hat Linux* ou si vous avez songé à une manière d'améliorer ce manuel, faites-nous part de vos commentaires. Signalez l'erreur dans Bugzilla (à l'adresse http://bugzilla.redhat.com/bugzilla) dans la section *rhl-rg*.

N'oubliez pas de mentionner la référence du manuel:

```
rhl-rg(FR)-9-Print-RHI (2003-02-13T19:20)
```

Nous pourront ainsi connaître la version du guide à laquelle vous faites référence.

Si vous avez la moindre suggestion susceptible d'améliorer la documentation, essayez d'en donner une description aussi détaillée que possible. Si vous avez détecté une erreur, veuillez inclure le numéro de section et une partie du texte qui l'entoure, de façon à ce que nous puissions la retrouver aisément.

7. Enregistrez-vous pour bénéficier de l'assistance

Si vous avez une édition de Red Hat Linux 9, n'oubliez pas de vous inscrire pour bénéficier des avantages auxquels vous avez droit en tant que client Red Hat.

Vous aurez droit à certains ou tous les avantages suivants, selon le produit Red Hat Linux que vous avez acheté:

- Support Red Hat — L'équipe d'assistance de Red Hat, Inc. répondra à vos questions sur l'installation.
- Red Hat Network — Mettez facilement à jour vos paquetages et recevez des nouvelles concernant la sécurité, personnalisées à votre système. Visitez http://rhn.redhat.com pour obtenir de plus amples informations.
- *Under the Brim: La E-Newsletter Red Hat* — Recevez chaque mois les dernières nouvelles et informations sur les produits directement de Red Hat.

Pour vous inscrire, rendez-vous à l'adresse: http://www.redhat.com/apps/activate/. Vous trouverez votre numéro d'identification de produit (Product ID) sur une carte noire, rouge et blanche dans votre emballage Red Hat Linux.

Pour en savoir plus sur l'assistance technique Red Hat Linux, consultez l'annexe *Assistance technique* dans le *Guide d'installation de Red Hat Linux*.

Merci d'avoir choisi Red Hat Linux et bonne chance!

L'équipe de documentation de Red Hat

I. Références au système

Afin de gérer le système aussi efficacement que possible, il est primordial de disposer de certaines connaissances sur ses composants et leur imbrication. Cette partie examine de nombreux aspects importants du système. Elle couvre le processus de démarrage, l'organisation de base d'un système de fichier, l'emplacement de fichiers système et de systèmes de fichiers essentiels et les concepts de base derrière les notions d'utilisateurs et de groupes. De plus, le système X Window fait l'objet d'un examen détaillé.

Table des matières



Chapitre 1. Processus de démarrage, Init et arrêt

Une des caractéristiques importantes de Red Hat Linux concerne la méthode - flexible et configurable par l'utilisateur - employée pour le démarrage de son système. Les utilisateurs peuvent configurer librement de nombreux aspects du processus de démarrage, y compris la possibilité de spécifier les programmes lancés au démarrage. De même, l'arrêt du système met fin nettement aux processus et ce, de manière organisée et configurable; bien que la personnalisation de ce processus ne soit que rarement nécessaire.

La compréhension des processus de démarrage et d'arrêt vous permettra non seulement de personnaliser facilement Red Hat Linux, mais également de résoudre plus rapidement les problèmes liés au démarrage ou à l'arrêt de votre système.

1.1. Processus de démarrage

Vous trouverez ci-dessous les étapes de base du processus de démarrage d'un système x86:

1. Le BIOS du système vérifie le système et lance le chargeur de démarrage de première étape sur le bloc de démarrage maître (MBR) du disque dur principal.
2. Le chargeur de démarrage de l'Étape 1 se charge en mémoire et lance le chargeur de démarrage de l'Étape 2 à partir de la partition `/boot/`.
3. Le chargeur de démarrage de l'Étape 2 charge le noyau en mémoire , qui à son tour, charge tout module nécessaire et monte la partition root en lecture-seule.
4. Le noyau passe le contrôle du processus de démarrage au programme `/sbin/init`.
5. Le programme `/sbin/init` charge tous les services et les outils de l'espace utilisateur et monte toutes les partitions répertoriées dans `/etc/fstab`.
6. L'utilisateur voit alors une invite de connexion pour le système Linux venant d'être démarré.

Étant donné que la configuration du processus de démarrage est plus commune que la personnalisation du processus d'arrêt, le reste de ce chapitre examinera en détail le fonctionnement du processus de démarrage et vous expliquera comment l'adapter à vos besoins spécifiques.

1.2. Examen détaillé du processus de démarrage

Le début du processus de démarrage varie en fonction de la plate-forme matérielle utilisée. Toutefois, une fois le noyau trouvé et chargé par le chargeur de démarrage, le processus de démarrage par défaut est identique pour toutes les architectures. Ce chapitre se concentre sur l'architecture x86.

1.2.1. Le BIOS

Lorsque l'on démarre un ordinateur x86, le processeur recherche le programme *BIOS* (de l'anglais '*Basic Input/Output System*') dans la mémoire morte (ROM) de la carte mère et l'exécute. Le BIOS est le plus bas niveau d'interface pour les périphériques et contrôle la première étape du processus de démarrage. Pour cette raison, le programme du BIOS est écrit en lecture seulement dans la mémoire morte et peut toujours être utilisé.

D'autres plates-formes utilisent différents programmes pour réaliser des tâches de bas niveau plus ou moins équivalentes à celles effectuées par le BIOS sur un système x86. Par exemple, les ordinateurs

Itanium utilisent le *Shell* '*Extensible Firmware Interface*' (ou *EFI*), tandis que les systèmes Alpha utilisent la *console SRM*.

Une fois chargé, le BIOS teste le système, recherche et vérifie les périphériques et trouve ensuite un périphérique valide qui sera utilisé pour amorcer le système. Normalement, il vérifie d'abord les lecteurs de disquettes et les lecteurs CD-ROM afin de trouver un support amorçable - s'il y en a un - puis se tourne vers les disques durs. L'ordre des unités recherchées lors du démarrage peut généralement être contrôlé par un paramètre du BIOS; il cherche sur le dispositif IDE maître sur le bus IDE principal. Le BIOS charge ensuite en mémoire tout programme résidant dans le premier secteur de ce dispositif, appelé le '*Master Boot Record*' (ou *MBR*). Le MBR ne fait que 512 octets et contient des instructions de codes pour démarrer la machine - appelée chargeur de démarrage - ainsi que la table de partitions. Une fois que le BIOS trouve et charge en mémoire le programme du chargeur de démarrage, il lui cède le contrôle du processus de démarrage.

1.2.2. Chargeur de démarrage

Cette section examine le processus de démarrage pour la plate-forme x86. Le processus de démarrage de votre ordinateur peut varier légèrement en fonction de son architecture. Reportez-vous à la Section 1.2.2.1 pour obtenir un bref aperçu des chargeurs de démarrage autres que ceux utilisés pour x86.

Sous Red Hat Linux, deux chargeurs de démarrage (aussi appelés chargeurs d'amorçage) sont disponibles: *GRUB* ou *LILO*. GRUB est le chargeur de démarrage par défaut, mais LILO est disponible pour ceux qui en ont besoin pour leur configuration matérielle, ou qui préfèrent l'utiliser. Pour de plus amples informations sur la configuration et l'utilisation de GRUB ou de LILO, reportez-vous au Chapitre 2.

Les deux chargeurs de démarrage pour la plate-forme x86 sont divisés au minimum en deux étapes. La première est un petit binaire de code machine. Son seul rôle est de localiser le chargeur de démarrage Étape 2 et d'en charger la première partie en mémoire.

GRUB est le chargeur de démarrage le plus récent qui a l'avantage de pouvoir lire les partitions ext2 et ext3 [1]et de charger son fichier de configuration — `/boot/grub/grub.conf`— au moment du démarrage. Pour de plus amples informations sur la façon de modifier ce fichier, reportez-vous à la Section 2.7.

Avec LILO, le chargeur de démarrage Étape 2 utilise des informations sur le MBR pour déterminer les options de démarrage dont dispose l'utilisateur. Cela signifie que chaque fois qu'un changement de configuration est réalisé ou que vous mettez manuellement à jour votre noyau, vous devez exécuter la commande `/sbin/lilo -v -v` pour écrire les informations appropriées sur le MBR. Pour plus de détails à ce propos, consultez la Section 2.8.


Astuce

Si vous mettez à niveau le noyau en utilisant l'application **Agent de mise à jour Red Hat**, le fichier de configuration du chargeur d'amorçage sera mis à jour automatiquement. Pour plus d'informations sur Red Hat Network, rendez-vous à l'adresse suivante: https://rhn.redhat.com.

Une fois que le chargeur de démarrage Étape 2 en mémoire, il affiche l'écran graphique initial Red Hat Linux indiquant à l'utilisateur les différents systèmes d'exploitation ou noyaux qu'il doit charger en fonction de sa configuration. Sur cet écran, l'utilisateur peut, à l'aide des touches flèchées, choisir le système d'exploitation ou le noyau qu'il souhaite charger et valider ce choix en pressant la touche

1. GRUB lit les systèmes de fichiers ext3 en tant que ext2, sans tenir compte du fichier journal. Reportez-vous au chapitre intitulé *Le système de fichiers ext3* du *Guide de personnalisation de Red Hat Linux* pour de plus amples informations sur le système de fichiers ext3.

[Entrée]. Si l'utilisateur n'appuie sur aucune touche avant qu'un certain laps de temps - configurable - ne ce soit écoulé, le chargeur de démarrage chargera la sélection par défaut.



Remarque

Si vous avez installé la prise en charge de noyau 'Symmetric Multi-Processor' (SMP), plusieurs options seront proposées la première fois que vous démarrerez votre système. Sous LILO, vous verrez `linux`, qui est le noyau SMP et `linux-up`, qui est pour des processeurs simples. Sous GRUB, vous verrez `Red Hat Linux (`<*version-noyau*>`-smp)`, qui est le noyau SMP et `Red Hat Linux (`<*version-noyau*>`)`, qui est pour des processeurs simples.

Si vous rencontrez des problèmes en utilisant le noyau SMP, sélectionnez le noyau non-stop au redémarrage.

Une fois que le chargeur de démarrage Étape 2 a déterminé le noyau à démarrer, il localise le binaire de noyau correspondant dans le répertoire `/boot/`. Le binaire du noyau est baptisé d'après le format— fichier `/boot/vmlinuz-`<*version-noyau*> (où <*version-noyau*> correspond à la version du noyau spécifiée dans les paramètres du chargeur de démarrage).

Pour obtenir des informations concernant l'utilisation du chargeur de démarrage pour transmettre des arguments de la ligne de commande au noyau, lisez le Chapitre 2. Pour des informations sur la manière de changer le niveau d'exécution à l'invite de GRUB ou LILO, lisez la Section 2.10.

Ensuite, le chargeur de démarrage place l'image *disque RAM initial* appropriée, appelée `initrd`, en mémoire. Cette image `initrd` est utilisée par le noyau pour charger les pilotes nécessaires au démarrage du système. Ceci s'avère particulièrement important si vous avez des disques durs SCSI ou si vous utilisez le système de fichiers ext3 [2].



Avertissement

Ne supprimez le répertoire `/initrd/` du système de fichiers sous aucun prétexte. Le retirer provoquerait un échec de votre système, avec un message d'erreur panique du noyau au moment du démarrage.

Une fois que le noyau et l'image *initrd* sont chargés en mémoire, le chargeur de démarrage cède le contrôle du processus de démarrage au noyau.

Pour obtenir un aperçu détaillé des chargeurs de démarrage GRUB et LILO, reportez-vous au Chapitre 2.

1.2.2.1. Chargeurs de démarrage pour d'autres architectures

Une fois que le noyau se charge et qu'il passe les commandes à `init`, les mêmes événements se produisent sur toutes les architectures. La différence essentielle entre le processus de démarrage de chaque architecture réside dans le choix de l'application utilisée pour trouver et charger le noyau.

Par exemple, l'architecture Alpha utilise le chargeur de démarrage `aboot`, tandis que l'architecture Itanium utilise le chargeur de démarrage ELILO.

Consultez le *Guide d'installation de Red Hat Linux* spécifique à ces plates-formes, pour obtenir de plus amples informations sur la manière de configurer leurs chargeurs de démarrage.

---

2. Pour obtenir des informations concernant la création d'un *initrd*, consultez le chapitre intitulé *Le système de fichiers ext3* du *Guide de personnalisation de Red Hat Linux*.

1.2.3. Le noyau

Lors du chargement du noyau, ce dernier initialise et configure immédiatement la mémoire de l'ordinateur. Ensuite, il configure les divers matériels attachés au système, y compris tous les processeurs et sous-systèmes E/S, ainsi que les périphériques de stockage. Il recherche ensuite l'image `initrd` compressée dans un emplacement prédéterminé dans la mémoire, la décompresse, la monte et charge tous les pilotes nécessaires. Ensuite, il initialise les dispositifs virtuels liés aux systèmes de fichiers, tels que LVM ou RAID logiciel, avant de démonter l'image disque `initrd` et de libérer par là-même toute la mémoire qu'elle occupait.

Le noyau crée alors un dispositif root, monte la partition root en lecture seule et libère la mémoire non-utilisée.

À ce stade, le noyau est chargé en mémoire et est désormais opérationnel. Toutefois, en l'absence de toute application offrant à l'utilisateur la possibilité de donner des informations utiles au système, on ne peut pas faire grand chose.

Afin de configurer l'environnement utilisateur, le noyau exécute de progamme `/sbin/init`.

1.2.4. Le programme `/sbin/init`

Le programme `/sbin/init` (aussi appelé `init`) coordonne le reste du processus de démarrage et configure l'environnement de l'utilisateur.

Lorsque la commande `init` est lancée, elle devient l'élément parent ou grand-parent de tous les processus qui sont lancés automatiquement sur votre système Red Hat Linux. Tout d'abord, elle exécute le script `/etc/rc.d/rc.sysinit` qui établit votre chemin d'accès d'environnement, démarre swap, vérifie les systèmes de fichiers, etc et s'occupe de tout ce qui doit être fait sur le système au moment de son initialisation. Par exemple, la plupart de systèmes utilisent une horloge, donc `rc.sysinit` lit le fichier de configuration `/etc/sysconfig/clock` sur ceux-ci pour initialiser l'horloge. Autre exemple: si vous avez des processus de port série spéciaux qui doivent être initialisés, `rc.sysinit` exécutera le fichier `/etc/rc.serial`.

La commande `init` exécute ensuite le script `/etc/inittab` qui décrit comment le système doit être configuré dans chaque *niveau d'exécution SysV init* [3]. Entre autres choses, le fichier `/etc/inittab` règle le niveau d'exécution par défaut et établit que `/sbin/update` doit s'exécuter chaque fois qu'il démarre un niveau d'exécution donné[4].

Ensuite, la commande `init` configure la bibliothèque de fonctions sources, `/etc/rc.d/init.d/functions`, pour le système. Celle-ci indique comment démarrer ou arrêter un programme et comment déterminer le PID d'un programme.

Le programme `init` démarre tous les processus d'arrière-plan en recherchant dans le répertoire `rc` approprié le niveau d'exécution spécifié comme niveau par défaut dans `/etc/inittab`. Les répertoires `rc` sont numérotés de façon à correspondre au niveau d'exécution qu'ils représentent. Par exemple, `/etc/rc.d/rc5.d/` est le répertoire correspondant au niveau d'exécution 5.

En démarrant au niveau d'exécution 5, le programme `init` examine le répertoire `/etc/rc.d/rc5.d/` afin de déterminer les processus à arrêter et à démarrer.

Ci-dessous figure un exemple de listing pour un répertoire `/etc/rc.d/rc5.d/`:

```
K05innd->../init.d/innd
K05saslauthd->../init.d/saslauthd
K10psacct->../init.d/psacct
K12cWnn->../init.d/cWnn
K12FreeWnn->../init.d/FreeWnn
K12kWnn->../init.d/kWnn
```

3. Pour plus d'informations sur SysV init, voir la Section 1.4.
4. La commande `update` est utilisée pour nettoyer les tampons sales.

```
K12mysqld->../init.d/mysqld
K12tWnn->../init.d/tWnn
K15httpd->../init.d/httpd
K15postgresql->../init.d/postgresql
K16rarpd->../init.d/rarpd
K20bootparamd->../init.d/bootparamd
K20iscsi->../init.d/iscsi
K20netdump-server->../init.d/netdump-server
K20nfs->../init.d/nfs
K20rstatd->../init.d/rstatd
K20rusersd->../init.d/rusersd
K20rwalld->../init.d/rwalld
K20rwhod->../init.d/rwhod
K24irda->../init.d/irda
K25squid->../init.d/squid
K28amd->../init.d/amd
K34dhcrelay->../init.d/dhcrelay
K34yppasswdd->../init.d/yppasswdd
K35atalk->../init.d/atalk
K35dhcpd->../init.d/dhcpd
K35smb->../init.d/smb
K35vncserver->../init.d/vncserver
K35winbind->../init.d/winbind
K40mars-nwe->../init.d/mars-nwe
K45arpwatch->../init.d/arpwatch
K45named->../init.d/named
K45smartd->../init.d/smartd
K46radvd->../init.d/radvd
K50netdump->../init.d/netdump
K50snmpd->../init.d/snmpd
K50snmptrapd->../init.d/snmptrapd
K50tux->../init.d/tux
K54pxe->../init.d/pxe
K55routed->../init.d/routed
K61ldap->../init.d/ldap
K65identd->../init.d/identd
K65kadmin->../init.d/kadmin
K65kprop->../init.d/kprop
K65krb524->../init.d/krb524
K65krb5kdc->../init.d/krb5kdc
K70aep1000->../init.d/aep1000
K70bcm5820->../init.d/bcm5820
K74ntpd->../init.d/ntpd
K74ups->../init.d/ups
K74ypserv->../init.d/ypserv
K74ypxfrd->../init.d/ypxfrd
K84bgpd->../init.d/bgpd
K84ospf6d->../init.d/ospf6d
K84ospfd->../init.d/ospfd
K84ripd->../init.d/ripd
K84ripngd->../init.d/ripngd
K85zebra->../init.d/zebra
K90isicom->../init.d/isicom
K92ipvsadm->../init.d/ipvsadm
K95firstboot->../init.d/firstboot
S00microcode_ctl->../init.d/microcode_ctl
S05kudzu->../init.d/kudzu
S08ip6tables->../init.d/ip6tables
```

```
S08ipchains->../init.d/ipchains
S08iptables->../init.d/iptables
S09isdn->../init.d/isdn
S10network->../init.d/network
S12syslog->../init.d/syslog
S13portmap->../init.d/portmap
S14nfslock->../init.d/nfslock
S17keytable->../init.d/keytable
S20random->../init.d/random
S24pcmcia->../init.d/pcmcia
S25netfs->../init.d/netfs
S26apmd->../init.d/apmd
S28autofs->../init.d/autofs
S44acpid->../init.d/acpid
S55sshd->../init.d/sshd
S56rawdevices->../init.d/rawdevices
S56xinetd->../init.d/xinetd
S80sendmail->../init.d/sendmail
S80spamassassin->../init.d/spamassassin
S84privoxy->../init.d/privoxy
S85gpm->../init.d/gpm
S90canna->../init.d/canna
S90crond->../init.d/crond
S90cups->../init.d/cups
S90xfs->../init.d/xfs
S95anacron->../init.d/anacron
S95atd->../init.d/atd
S97rhnsd->../init.d/rhnsd
S99local->../rc.local
S99mdmonitor->../init.d/mdmonitor
```

Comme le montre ce listing, aucun des scripts qui lancent et arrêtent vraiment les services n'est réellement situé dans le répertoire `/etc/rc.d/rc5.d/`. Tous les fichiers dans `/etc/rc.d/rc5.d/` sont en fait des *liens symboliques* qui pointent vers les scripts situés dans le répertoire `/etc/rc.d/init.d/`. Des liens symboliques sont utilisés dans chacun des répertoires `rc` afin que les niveaux d'exécution puissent être reconfigurés en créant, modifiant et supprimant les liens symboliques, et ce, sans affecter les scripts auxquels ils font référence.

Le nom de chaque lien symbolique commence par `K` ou `S`. Les liens `K` correspondent à des processus arrêtés à ce niveau d'exécution, tandis que les liens `S` correspondent à des processus démarrés.

La commande `init` arrête tout d'abord tous les liens symboliques `K` du répertoire en émettant la commande `/etc/rc.d/init.d/`<*commande*>` stop`, <*commande*> correspondant au processus à arrêter. Elle démarre ensuite tous les liens symboliques `S` en émettant la commande `/etc/rc.d/init.d/`<*commande*>` start`.



Astuce

> Une fois que le système a terminé son démarrage, il est possible d'établir une connexion en tant que super-utilisateur et d'exécuter ces mêmes scripts pour arrêter et démarrer des services. Par exemple, la commande `/etc/rc.d/init.d/httpd stop` arrêtera le serveur Web Apache.

Chacun des liens symboliques est numéroté de façon à établir l'ordre de démarrage. L'ordre dans lequel les services sont démarrés ou arrêtés peut être modifié en changeant ce numéro. Plus le numéro est bas, plus avancée sera la place dans l'ordre de démarrage. Les liens symboliques disposant du même numéro sont démarrés par ordre alphabétique.



Remarque

> Une des dernières choses que le programme `init` exécute est le fichier `/etc/rc.d/rc.local`. Ce dernier est utilise pour la personnalisation du système. Reportez-vous à la Section 1.3 pour de plus amples informations sur l'utilisation du fichier `rc.local`.

Une fois que la commande `init` a progressé dans le répertoire `rc` approprié pour le niveau d'exécution, le script `/etc/inittab` établit un processus `/sbin/mingetty` pour chaque console virtuelle (invites de login) assignée à ce niveau d'exécution. Les niveaux d'exécution 2 à 5 obtiennent tous six consoles virtuelles, tandis que le niveau d'exécution 1 (mode mono-utilisateur) n'obtient qu'une console et que les niveaux d'exécution 0 et 6 n'en obtiennent aucune. Le processus `/sbin/mingetty` ouvre des lignes de communication vers les dispositifs *tty* [5], règle leurs modes, imprime l'invite de login, prend le nom d'utilisateur, puis commence le processus de login pour l'utilisateur concerné.

Au niveau d'exécution 5, `/etc/inittab` exécute un script appelé `/etc/X11/prefdm`. Le script `prefdm` exécute le gestionnaire d'affichage X préféré — `gdm`, `kdm`, ou `xdm`, en fonction de ce qui est contenu dans le fichier `/etc/sysconfig/desktop`.

À ce stade, le système devrait fonctionner à un niveau d'exécution 5 et devrait afficher une invite de connexion à l'écran.

1.3. Exécution de programmes supplémentaires au démarrage

Le script `/etc/rc.d/rc.local` est exécuté par la commande `init` au démarrage ou lors de la modification des niveaux d'exécution. L'ajout de commandes à ce script est une façon simple d'exécuter des tâches nécessaires comme le démarrage de services spéciaux ou l'initialisation de périphériques sans devoir écrire des scripts d'initialisation compliqués dans le répertoire `/etc/rc.d/init.d/` et créer des liens symboliques.

Le script `/etc/rc.serial` est utilisé si des ports série doivent être configurés au démarrage. Ce script exécute les commandes `setserial` pour la configuration des port série du système. Consultez les pages de manuel relatives à `setserial` pour obtenir de plus amples informations.

1.4. Niveaux d'exécution de SysV Init

Le système de niveaux d'exécution SysV init fournit un processus standard pour contrôler les programmes lancés et arrêtés par `init` lors de l'initialisation d'un niveau d'exécution. SysV init a été choisi parce qu'il est non seulement plus facile à utiliser et mais également plus flexible que le processus init BSD traditionnel.

Les fichiers de configuration de SysV init se trouvent dans le répertoire `/etc/rc.d/`. Dans ce répertoire, se trouvent les scripts `rc`, `rc.local`, `rc.sysinit` et, de manière optionnelle, les scripts `rc.serial` ainsi que les répertoires suivants:

```
init.d/
rc0.d/
rc1.d/
rc2.d/
rc3.d/
rc4.d/
rc5.d/
rc6.d/
```

5. Consultez la Section 5.3.11 pour des informations supplémentaires sur les périphériques `tty`.

Le répertoire init.d/ contient les scripts utilisés par la commande /sbin/init pour le contrôle des services. Chacun des répertoires numérotés représentent les six niveaux d'exécution configurés par défaut sous Red Hat Linux.

1.4.1. Niveaux d'exécution (Runlevels)

Les niveaux d'exécution correspondent à un état, ou *mode*, défini par les services dans le répertoire /etc/rc.d/rc<*x*>.d/ de SysV, où <*x*> représente le numéro du niveau d'exécution.

L'idée derrière les niveaux d'exécution de SysV init se résume au principe que divers systèmes peuvent être utilisés de différentes manières. Par exemple, un serveur fonctionne plus efficacement lorsqu'il n'est pas dépendant de l'utilisation des ressources du système par le système X Window. En d'autres occasions, il se peut qu'un administrateur système doive faire fonctionner le système à un niveau d'exécution inférieur afin d'effectuer des tâches de diagnostic; comme par exemple pour résoudre la corruption de disques à un niveau d'exécution 1, lorsque les utilisateurs n'utilisent pas le système.

Les caractéristiques d'un niveau d'exécution donné déterminent les services qui seront arrêtés ou démarrés par init. Par exemple, le niveau d'exécution 1 (mode mono-utilisateur) arrête tout service réseau alors que le niveau d'exécution 3 lui, démarre ces mêmes services. En déterminant le démarrage ou l'arrêt de services spécifiques à un niveau d'exécution donné, init peut rapidement changer le mode de l'ordinateur sans que l'utilisateur n'ait à arrêter ou démarrer ces services manuellement.

Les niveaux d'exécution suivants sont définis par défaut pour Red Hat Linux:

- 0 — Arrêt
- 1 — Mode texte mono-utilisateur
- 2 — Pas utilisé
- 3 — Mode texte multi-utilisateurs complet
- 4 — Pas utilisé
- 5 — Mode graphique multi-utilisateurs complet (avec un écran de connexion de type X Window)
- 6 — Redémarrage

En général, les utilisateurs font fonctionner Red Hat Linux à un niveau d'exécution 3 ou 5 — les deux niveaux correspondant à des modes multi-utilisateurs complets. Parfois, les utilisateurs personnalisent les niveaux d'exécution 2 et 4 pour leurs besoins spécifiques, puisque ces derniers ne sont pas utilisés.

Le niveau d'exécution par défaut du système se trouve dans /etc/inittab. Pour trouver le niveau d'exécution par défaut d'un système, recherchez la ligne, semblable à celle reproduite ci-dessous, au début de /etc/inittab:

```
id:5:initdefault:
```

Dans l'exemple ci-dessus, le niveau d'exécution par défaut est 5, comme l'indique le chiffre qui suit le premier signes des deux-points (:). Si vous désirez le changer, modifiez /etc/inittab en étant connecté en tant que super-utilisateur.



Faites très attention lorsque vous éditez /etc/inittab. De simples fautes de frappe peuvent empêcher votre système de démarrer. Si cela se produit, vous devrez utiliser une disquette d'amorçage pour votre système ou passer en mode mono-utilisateur ou en mode de secours pour redémarrer l'ordinateur et réparer le fichier.

Pour plus d'informations sur le mode mono-utilisateur ou le mode de secours, reportez-vous au chapitre intitulé *Mode de secours* du *Guide de personnalisation de Red Hat Linux*.

Il est possible de changer le niveau d'exécution par défaut au moment du démarrage en modifiant les arguments transmis par le chargeur de démarrage au noyau. Pour toute information sur la modification du niveau d'exécution au démarrage, reportez-vous à la Section 2.10.

1.4.2. Utilitaires de niveaux d'exécution

Une des meilleures façons de configurer les niveaux d'exécution consiste à utiliser un des *utilitaires initscript*. Ces outils sont conçus pour simplifier le maintien des fichiers dans la hiérarchie du répertoire SysV init et pour éviter aux administrateurs système de manipuler directement les nombreux liens symboliques des sous-répertoires `/etc/rc.d/`.

Red Hat Linux offrent trois utilitaires de ce type:

- `/sbin/chkconfig` — l'utilitaire `/sbin/chkconfig` est un outil de ligne de commande simple permettant de maintenir la hiérarchie des répertoires `/etc/rc.d/init.d`.
- **/sbin/ntsysv** — l'utilitaire **/sbin/ntsysv** basé sur ncurses fournit une interface interactive de mode texte, que certains utilisateurs trouvent plus simple à utiliser que `chkconfig`.
- L'**Outil de configuration des services** — le programme graphique **Outil de configuration des services** (`redhat-config-services`) est un utilitaire flexible basé sur GTK2 permettant de configurer les niveaux d'exécution.

Veuillez vous reporter au chapitre concernant le *Contrôle de l'accès aux services* du *Guide de personnalisation de Red Hat Linux* pour obtenir de plus amples informations sur ces outils.

1.5. Arrêt

Pour arrêter Red Hat Linux, le super-utilisateur peut exécuter la commande `/sbin/shutdown`. La page de manuel relative à `shutdown` contient une liste complète des options; ceci étant, les deux options les plus courantes sont les suivantes:

```
/sbin/shutdown-hnow
/sbin/shutdown-rnow
```

Après avoir tout arrêté, l'option `-h` éteindra l'ordinateur et l'option `-r` le redémarrera.

Les utilisateurs autres que les super-utilisateur peuvent utiliser les commandes `reboot` et `halt` pour éteindre l'ordinateur en étant à un niveau d´exécution entre 1 et 5. Tous les systèmes d'exploitation Linux ne prennent cependant pas en charge cette fonction.

Si l'ordinateur ne s'éteint pas automatiquement, ne le faites pas manuellement avant que le message confirmant l'arrêt du système n'apparaisse à l'écran.

Si vous n'attendez pas ce message, il se peut que toutes les partitions du disque dur n'aient pas été complètement démontées, ce qui pourrait entraîner la corruption de systèmes de fichiers.



Chapitre 2. Chargeurs de démarrage

Avant que Red Hat Linux ne puisse s'exécuter sur un système, il doit être chargé en mémoire par un progamme spécial appelé *chargeur de démarrage*. Un chargeur de démarrage existe généralement sur le disque dur principal du système (ou sur d'autres supports) et a pour seule responsabilité de charger en mémoire le noyau Linux ainsi que les fichiers dont il a besoin, ou (dans certains cas) d'autres systèmes d'exploitation.

2.1. Chargeurs de démarrage et architecture système

Chaque architecture système pouvant exécuter Red Hat Linux utilise un chargeur de démarrage différent. Par exemple, l'architecture Alpha utilise le chargeur de démarrage `aboot` tandis que l'architecture Itanium utilise le chargeur de démarrage `ELILO`.

Ce chapitre examine les commandes et options de configuration des deux chargeurs de démarrage fournis avec Red Hat Linux pour l'architecture x86, à savoir GRUB et LILO.

2.2. GRUB

Le *'GNU GRand Unified Boot loader'* ou GRUB est un programme permettant à l'utilisateur de sélectionner le système d'exploitation ou noyau installés à charger au démarrage du système. Il permet également à l'utilisateur de transmettre des arguments au noyau.

2.2.1. GRUB et le processus de démarrage x86

Cette section examine de façon plus détaillée le rôle spécifique que GRUB joue lors du démarrage d'un système x86. Pour obtenir un aperçu du processus de démarrage global, reportez-vous à la Section 1.2.

GRUB se charge en mémoire en suivant les étapes suivantes:

1. *Le chargeur de démarrage Étape 1 ou primaire, est lu en mémoire par le BIOS à partir du MBR*[1]. Le chargeur de démarrage primaire existe sur moins de 512 octets d'espace disque dans le MBR et peut charger aussi bien le chargeur de démarrage Étape 1.5 que le chargeur de démarrage Étape 2.
2. *Le chargeur de démarrage Étape 1.5 est lu en mémoire par le chargeur de démarrage, si cela est nécessaire.* Certains matériels requièrent une étape intermédiaire pour arriver au chargeur de démarrage Étape 2. Ceci peut être le cas si la partition `/boot` se situe au-dessus de la tête de cylindre 1024 du disque dur ou lorsque le mode LBA est utilisé. Le chargeur de démarrage Étape 1.5 se trouve sur la partition `/boot` ou sur une petite portion du MBR et de la partition `/boot`.
3. *Le chargeur de démarrage Étape 2 ou secondaire est lu en mémoire.* Le chargeur de démarrage secondaire affiche le menu et l'environnement de commandes GRUB. Cette interface vous permet de sélectionner le système d'exploitation ou noyau Linux à démarrer, de transférer des arguments au noyau ou de vérifier des paramètres du systèmes, comme la quantité de mémoire (RAM) disponible.
4. *Le chargeur de démarrage secondaire lit le système d'exploitation ou noyau et `initrd` en mémoire.* Une fois que GRUB détermine le système d'exploitation à démarrer, il le charge en mémoire et cède le contrôle de la machine à ce système d'exploitation.

1. Pour en savoir plus sur le BIOS et le MBR, voir la Section 1.2.1.

La méthode de démarrage utilisée pour charger Red Hat Linux est appelée la méthode de *chargement direct* car le chargeur de démarrage charge directement le système d'exploitation. Il n'y a pas d'intermédiaire entre le chargeur de démarrage et le noyau.

Le processus de démarrage utilisé par d'autres systèmes d'exploitation peut différer. Par exemple, les systèmes d'exploitation DOS et Windows de Microsoft, ainsi que divers autres systèmes d'exploitation propriétaires, utilisent une méthode de démarrage basée sur le *chargement en chaîne*. Avec cette méthode, le MBR pointe simplement vers le premier secteur de la partition contenant le système d'exploitation. A cet endroit, il trouve les fichiers permettant de démarrer véritablement ce système d'exploitation.

GRUB prend en charge les méthodes de chargement direct et en chaîne, ce qui permet au système de fonctionner sur la quasi totalité des systèmes d'exploitation.



Avertissement

Lors de l'installation, le programme d'installation DOS et Windows de Microsoft écrase complètement le MBR, détruisant par là-même tout chargeur de démarrage existant. Si vous créez un système de démarrage double, nous vous conseillons d'installer en premier le système d'exploitation Microsoft. Pour obtenir des instructions à ce propos, reportez-vous à l'appendice intitulée *Installation de Red Hat Linux dans un environnement de démarrage double* du *Guide d'installation de Red Hat Linux*.

2.2.2. Caractéristiques de GRUB

GRUB contient un certain nombre de caractéristiques qui le rendent plus intéressant que d'autres chargeurs de démarrage disponibles pour l'architecture x86. Vous trouverez ci-dessous une liste de certaines des caractéristiques les plus importantes:

- *GRUB offre un véritable environnement pré-système d'exploitation basé sur les commandes utilisées sur les ordinateurs x86.* Ceci permet à l'utilisateur de bénéficier d'une flexibilité maximale pour le chargement de systèmes d'exploitation avec certaines options ou pour obtenir des informations sur le système. De nombreuses architectures autres que l'architecture x86 ont utilisé pendant des années des environnements pré-système d'exploitation permettant de contrôler le mode de démarrage depuis une ligne de commande. Bien que LILO et d'autres chargeurs de démarrage x86 offrent certaines fonctionnalités de commande, GRUB est doté d'un éventail de fonctions plus large.
- *GRUB prend en charge le mode 'Logical Block Addressing' (LBA).* Le mode LBA place les conversions d'adressage utilisées pour localiser des fichiers dans le micrologiciel du disque et est utilisé sur de nombreux dispositifs IDE et sur tous les dispositifs SCSI. Avant l'arrivée du mode LBA, les chargeurs de démarrage pouvaient se heurter à la limite BIOS de 1024 cylindres, créant des situations dans lesquelles le BIOS se trouvait dans l'incapacité de trouver des fichiers au-delà de cette tête de cylindre du disque. La prise en charge du mode LBA permet à GRUB de procéder à l'amorçage de systèmes d'exploitation résidant sur des partitions situées au-delà de la limite des 1024 cylindres, à condition que votre BIOS prenne en charge le mode LBA. La plupart des révisions BIOS modernes prennent en charge le mode LBA.
- *GRUB peut lire les partitions ext2.* Cette fonctionnalité permet à GRUB d'accéder à son fichier de configuration, `/boot/grub/grub.conf`, chaque fois que le système démarre, évitant ainsi à l'utilisateur d'écrire une nouvelle version du chargeur de démarrage première étape sur le MBR lors de toute modification de la configuration. L'utilisateur ne devra réinstaller GRUB sur le MBR que si l'emplacement physique de la partition `/boot` est déplacé sur le disque. Pour en savoir plus sur l'installation de GRUB sur le MBR, reportez-vous à la Section 2.3.

2.3. Installation de GRUB

Si, pendant le processus d'installation de Red Hat Linux, GRUB n'était pas installé, vous pouvez l'installer ultérieurement. Une fois installé, il devient automatiquement le chargeur de démarrage par défaut.

Avant d'installer GRUB, vérifiez que vous disposez du paquetage GRUB le plus récent, ou utilisez le paquetage GRUB des CD-ROM d'installation Red Hat Linux. Pour obtenir des instructions sur l'installation de paquetages, reportez-vous au chapitre intitulé *Gestion des paquetages avec RPM* du *Guide de personnalisation de Red Hat Linux*.

Une fois le paquetage GRUB installé, ouvrez une invite de shell root et lancez la commande /sbin/grub-install <*emplacement*>, où <*emplacement*> correspond à l'emplacement où le chargeur de démarrage GRUB Étape 1 doit être installé.

La commande qui suit installe GRUB sur le MBR du dispositif IDE maître sur le bus IDE primaire: /sbin/grub-install /dev/hda

Lors du prochain démarrage de votre système, le menu chargeur de démarrage graphique de GRUB apparaîtra avant le chargement du noyau en mémoire.

2.4. Terminologie relative à GRUB

Un des points fondamentaux à maîtriser avant d'utiliser GRUB est la façon dont le programme fait référence aux périphériques, tels que votre disque dur et les partitions. Cette information est très importante lorsque vous configurez GRUB pour lui permettre le démarrage de plusieurs systèmes d'exploitation.

2.4.1. Noms des périphériques

Supposons par exemple qu'un système ait plus d'un disque dur. Le premier disque dur d'un système est appelé (hd0) par GRUB. La première partition de ce disque est appelée (hd0,0) et la cinquième partition du second disque est appelée (hd1,4). En général, les règles de nomination pour les systèmes de fichiers, lorsque l'on utilise GRUB, se présentent comme suit:

(<*type-de-périphérique*><*numéro-périphérique-bios*>,<*numéro-partition*>)

Les parenthèses et la virgule sont très importantes dans les conventions de désignation des périphériques. L'élément <*type-de-périphérique*> se rapporte au périphérique spécifié: disque dur (hd) ou ou disquette (fd).

L'élément <*numéro-périphérique-bios*> est le numéro du périphérique en fonction du BIOS du système, en commençant à 0. Le disque dur IDE primaire est numéroté 0, alors que le disque dur IDE secondaire est numéroté 1. La façon dont l'ordre est établi est très proche de la façon dont le noyau de Linux dispose les périphériques avec des lettres. Là où la lettre a dans hda se rapporte à 0, la lettre b dans hdb se rapporte à 1 et ainsi de suite.



Remarque

Le système de numérotation de GRUB pour les périphériques commence par 0 et non pas 1. Le non respect de cette distinction est la source des erreurs les plus courantes commises par les nouveaux utilisateurs GRUB.

L'élément <*numéro-partition*> se rapporte au numéro d'une partition spécifique sur un périphérique disque. Comme pour l'élément <*numéro-de-périphérique-bios*>, la numérotation des

partitions commence par `0`. Même si la plupart des partitions sont désignées par des numéros, si votre système utilise des partitions BSD, celles-ci seront désignées par des lettres, comme `a` ou `c`.

GRUB fait appel aux règles suivantes pour désigner des périphériques et partitions:

- Peu importe si votre disque dur est IDE ou SCSI. Le nom de tous les disques durs commence par `hd`. Les lecteurs de disquette quant à eux commencent par `fd`.
- Pour indiquer un périphérique en entier sans spécifier ses partitions il suffit de retirer la virgule et le numéro de la partition. Ceci est important lorsque l'on souhaite que GRUB configure le bloc de démarrage maître pour un disque donné. Par exemple, `(hd0)` indique le MBR sur le premier périphérique et `(hd3)` indique le MBR sur le quatrième.
- Si vous possédez plusieurs disques durs, il est très important de connaître l'ordre de démarrage défini dans le BIOS. Cela reste assez simple à faire si vous ne possédez que des disques durs IDE ou SCSI, mais dès l'instant où tous les deux sont installés, les choses deviennent un peu plus compliquées.

2.4.2. Noms de fichiers et listes des blocs

En saisissant des commandes pour GRUB qui impliquent un fichier, comme une liste de menu qui permet le démarrage de plusieurs systèmes d'exploitation, il est impératif d'inclure le fichier immédiatement après avoir désigné le périphérique et la partition.

Un exemple de spécification pour un nom de fichier absolu se présente sous le format suivant:

(<*type-de-périphérique*><*numéro-périphérique-bios*>,<*numéro-partition*>)/*chemin-d'accès/ vers/fichier*

La plupart du temps, un utilisateur indiquera des fichiers en spécifiant le chemin d'accès sur cette partition plus le nom du fichier.

Vous pouvez également indiquer à GRUB des fichiers qui n'apparaissent pas dans le système de fichiers, tel qu'un chargeur de chaîne par exemple qui apparaît dans les tous premiers blocs d'une partition. Pour indiquer ces fichiers, vous devez fournir une *liste de blocs* qui explique à GRUB, bloc par bloc, l'emplacement du fichier sur la partition. Étant donné qu'un fichier peut être constitué de plusieurs blocs, il existe une manière particulière d'écrire une liste de blocs. Chaque emplacement de section de fichier est décrit par un numéro de bloc décalé, suivi d'un nombre de blocs après ce point de décalage; les sections sont reliées entre elles dans un ordre défini selon les virgules placées entre les différents éléments.

Prenons l'exemple de la liste de blocs suivante pour illustrer cette notion:

```
0+50,100+25,200+1
```

Cette liste de blocs indique à GRUB qu'il doit utiliser un fichier commençant au premier bloc de la partition et qui utilise les blocs 0 à 49, 99 à 124, et 199.

Savoir comment écrire des listes de blocs est très utile lorsque GRUB doit charger des systèmes d'exploitation qui utilisent le chargement de chaîne, comme Microsoft Windows. Vous pouvez laisser tomber le décalage de bloc si vous commencez au bloc 0. Par exemple, le fichier de chargement de chaîne dans la première partition du premier disque dur devrait s'appeler ainsi:

```
(hd0,0)+1
```

Vous pouvez également utiliser la commande `chainloader` suivante avec un mode d'indication de liste de blocs similaire à la ligne de commande GRUB après avoir spécifié le bon périphérique et la bonne partition et en étant connecté en tant que root:

```
chainloader +1
```

2.4.3. Système de fichiers root de GRUB

Certains utilisateurs sont désorientés par l'emploi du terme 'système de fichiers root' dans GRUB. Il est important de se rappeler que le système de fichiers root de GRUB n'a rien à voir avec le système de fichiers root de Linux.

Par système de fichiers root de GRUB on désigne la partition racine d'un périphérique donné. GRUB exploite notamment ces informations pour monter le périphérique et procéder au chargement des fichiers.

Avec Red Hat Linux, une fois que GRUB a chargé sa propre partition root (qui est l'équivalent de la partition `/boot`et contient le noyau Linux), la commande `kernel` peut être exécutée, avec l'emplacement du fichier de noyau en option. Lorsque le noyau Linux démarre, il établit le système de fichiers Linux auquel les utilisateurs sont habitué. Le système de fichiers root de GRUB et ses montages sont oubliés; ils ne servaient qu'au démarrage du fichier du noyau.

Pour de plus amples informations, lisez les notes relatives aux commandes `root` et `kernel` contenues dans la Section 2.6.

2.5. Interfaces GRUB

GRUB présente trois interfaces, qui fournissent différents niveaux de fonctionnalités. Chacune de ces interfaces permet aux utilisateurs de démarrer le noyau Linux ou d'autres systèmes d'exploitation.

Les interfaces sont les suivantes:

Interface Menu

Si la configuration de GRUB a été réalisée automatiquement par le programme d'installation de Red Hat Linux, ce sera l'interface affichée par défaut. Un menu des différents systèmes d'exploitation et noyaux pré-configurés avec leurs propres commandes de démarrage est présenté sous la forme de liste, organisée de façon nominale. Utilisez les flèches du clavier pour choisir une option différente de celle qui est présentée par défaut puis appuyez sur la touche [Entrée] pour valider la sélection. Si aucun choix n'est fait avant l'expiration d'un délai préétabli, GRUB procède au démarrage de l'option par défaut.

Appuyez sur la touche [e] pour accéder à l'interface éditeur d'entrées ou sur la touche [c] pour charger une interface de ligne de commande.

Pour plus d'informations sur la configuration de cette interface, lisez la Section 2.7.

Interface éditeur d'entrée de menu

Pour accéder à l'éditeur d'entrée de menu, appuyez sur la touche [e] depuis le menu du chargeur de démarrage. Les commandes de GRUB relatives à cette entrée sont présentées ci-après. Ces lignes de commande peuvent être modifiées par les utilisateurs avant le démarrage du système d'exploitation en ajoutant une ligne de commande ([o] insère la nouvelle ligne après la ligne actuelle et [O] l'insère avant), en en modifiant une ([e]) ou finalement en en supprimant une ([d]).

Une fois que vos modifications sont effectuées, appuyez sur la touche [b] pour les exécuter les commande et démarrer le système d'exploitation. La touche [Échap] elle, permet d'annuler ces modifications et recharge l'interface menu standard. Finalement, la touche [c] elle, charge l'interface de la ligne de commande.



Astuce

Pour de plus amples informations sur la façon de procéder pour changer les niveaux d'exécution avec GRUB en utilisant l'éditeur d'entrée de menu, reportez-vous à la Section 2.10.

Interface de ligne de commande

L'interface de ligne de commande, bien qu'étant la plus élémentaire des interfaces GRUB, est celle qui vous offre le plus de contrôle. La ligne de commande permet de taper toute commande GRUB pertinente et de l'exécuter en appuyant sur la touche [Entrée]. Cette interface présente certaines fonctions avancées ressemblant aux fonctions du shell comme, par exemple, la touche [Tab] pour l'achèvement automatique de ligne en fonction du contexte et les combinaisons de touches avec [Ctrl] lors de la saisie de commande, comme par exemple, [Ctrl]-[a] pour retourner au début de la ligne et [Ctrl]-[e] pour aller directement à la fin de la ligne. De plus, les touches de direction, [Début], [Fin] et [Suppr] fonctionnent de la même façon que sous le shell `bash`.

Pour obtenir une liste des commandes les plus courantes, reportez-vous à la Section 2.6.

2.5.1. Ordre des interfaces

Lorsque l'environnement GRUB charge le chargeur de démarrage secondaire, il part à la recherche de son fichier de configuration. Une fois que celui-ci a été trouvé, il l'utilise pour la construction de la liste de menu et affiche l'interface menu.

Si le fichier de configuration est introuvable ou s'il s'avère impossible à lire, GRUB charge l'interface de ligne de commande permettant à l'utilisateur de saisir manuellement les commandes nécessaires pour achever le processus de démarrage.

Si le fichier de configuration n'est pas valide, GRUB affiche l'erreur et attend une commande. Ceci aide l'utilisateur à déterminer exactement là où les problème est survenu. Appuyez sur une touche quelconque pour recharger l'interface menu d'où il est alors possible d'éditer l'option du menu et d'apporter les corrections nécessaires en fonction de l'erreur rapportée par GRUB. Si la correction apportée ne résout pas le problème, GRUB rapporte une erreur et charge de nouveau l'interface menu.

2.6. Les commandes GRUB

GRUB permet un certain nombre de commandes utiles dans son interface ligne de commande. Certaines de ces commandes acceptent une option après leur nom. Pour être acceptées, ces options doivent être séparées de la commande et des autres options présentes par un espace.

Ci-après figure une liste de commandes utiles:

- `boot` — démarre le système d'exploitation ou le chargeur de chaîne qui a été sélectionné et chargé précédemment.
- `chainloader` <*nom-de-fichier*> — charge le fichier indiqué comme chargeur de chaîne. Pour s'assurer que ce fichier sera pris dès le premier secteur de la première partition, utilisez `+1` comme nom de fichier.
- `displaymem` — affiche l'utilisation actuelle de mémoire, sur la base des informations fournies par le BIOS. Cette commande est pratique quand vous ignorez la quantité de mémoire vive dont le système dispose, avant de le démarrer.
- `initrd` <*nom-de-fichier*> — permet à l'utilisateur de spécifier un disque RAM initial à utiliser pour l'amorçage. Un `initrd` est nécessaire au noyau lorsque celui-ci a besoin de certains modules pour démarrer correctement, comme lorsque la partition root est formatée avec le système de fichiers ext3.

- `install` <*étape-1*> <*installer-disque*> <*étape-2*> `p` <*fichier-config*> — installe GRUB dans le bloc de démarrage maître (MBR) du système.

  Lors de l'utilisation de la commande `install`, il est nécessaire de spécifier les éléments suivants:

  - <*étape-1*> — précise un périphérique, ne partition et un fichier où l'image du premier chargeur de démarrage peut être trouvée, tel que `(hd0,0)/grub/stage1`.
  - <*installer-disque*> — spécifie le disque où le chargeur de démarrage de l'Étape 1 doit être installé, comme par exemple `(hd0)`.
  - <*étape-2*> — indique au chargeur de démarrage de l'Étape 1, l'emplacement du chargeur de démarrage de l'Étape 2 comme, par exemple, `(hd0,0)/grub/stage2`.
  - `p` <*fichier-config*> — cette option indique à la commande `install` de rechercher le fichier de configuration du menu spécifié par <*fichier-config*>. Un exemple de chemin d'accès valide au fichier de configuration est `(hd0,0)/grub/grub.conf`.



Avertissement

La commande `install` écrasera toute autre information sur le bloc de démarrage maître (MBR). Lors de son exécution, toutes les informations (autres que celles de GRUB) utilisées pour démarrer d'autres systèmes d'exploitation seront perdues.

- `kernel` <*nom-de-fichier-du-noyau*> <*option-1*> <*option-N*> — indique quel fichier du noyau charger depuis le système de fichiers root de GRUB, lors d'un chargement direct du système d'exploitation. La commande `kernel` peut être accompagnée d'options qui seront passées au noyau lors de son chargement.

  Pour Red Hat Linux, un exemple de commande `kernel` ressemble à l'extrait suivant:

  ```
  kernel /vmlinuz root=/dev/hda5
  ```

  Cette ligne indique que le fichier `vmlinuz` est chargé depuis le système de fichiers root de GRUB, tel que `(hd0,0)`. Une option est aussi passée au noyau indiquant que lors du chargement du système de fichiers root pour le noyau Linux, ce dernier doit se situer sur `hda5`, la cinquième partition du premier disque dur IDE. Plusieurs autres options peuvent être placées après cette option si nécessaire.

- `root` <*périphérique-et-partition*> — configure la partition racine (root) de GRUB pour en faire un périphérique et une partition spécifiques, comme par exemple `(hd0,0)`, et monte la partition afin que les fichiers puissent être lus.
- `rootnoverify` <*périphérique-et-partition*> — a les mêmes fonctions que la commande `root` mais ne monte pas la partition.

Il existe bien d'autres commandes. Pour obtenir une liste complète de ces dernières, tapez `info grub`.

2.7. Fichier de configuration du menu de GRUB

Le fichier de configuration (`/boot/grub/grub.conf`), utilisé pour créer la liste des systèmes d'exploitation à démarrer dans l'interface menu, permet à l'utilisateur de sélectionner un groupe préétabli de commandes à exécuter. Les commandes fournies dans la Section 2.6 peuvent être utilisées, ainsi que certaines commandes spéciales qui ne sont disponibles que dans le fichier de configuration.

2.7.1. Commandes spéciales du fichier de configuration

Les commandes suivantes ne peuvent être utilisées qu'avec le fichier de configuration du menu de GRUB:

- `color` <*couleur-normale*> <*couleur-sélectionnée*> — permet de définir les couleurs à utiliser dans le menu, soit une couleur pour le premier plan et une pour l'arrière-plan. Il est possible de n'utiliser que les noms de ces couleurs, comme `red/black` (rouge/noir) par exemple: `color red/black green/blue`
- `default` <*nom-titre*> — le titre de l'entrée par défaut qui sera chargée si le délai imparti pour le choix d'une option du menu est dépassé.
- `fallback` <*nom-titre*> — cette commande, par son utilisation, permet d'indiquer le titre de l'entrée à essayer dans le cas où la première tentative échoue.
- `hiddenmenu` — son utilisation empêche l'affichage de l'interface menu de GRUB, chargeant l'entrée par défaut (`default`) lorsque le durée d'attente initiale (`timeout`) est dépassée. L'utilisateur peut visualiser le menu standard de GRUB en appuyant sur la touche [Échap].
- `password` <*mot-de-passe*> — l'utilisation de cette commande permet d'interdire à tout utilisateur ne connaissant pas le mot de passe, d'éditer les entrées relatives à l'option de ce menu.

  Il est possible éventuellement, d'indiquer un autre fichier de configuration de menu après la commande `password`<*mot-de-passe*>. Dans ce cas, GRUB redémarrera le chargeur de démarrage Étape 2 et utilisera le deuxième fichier de configuration spécifié pour construire le menu. Si ce fichier alternatif n'est pas indiqué dans cette commande, tout utilisateur en possession du mot de passe sera à même d'éditer le fichier de configuration actuel.
- `timeout` — l'utilisation de cette commande permet de régler la durée, en secondes, qui peut s'écouler avant que GRUB ne charge l'entrée indiquée dans la commande `default`.
- `splashimage` — précise l'emplacement de l'image de fond utilisée lors du démarrage de GRUB.
- `title` — définit le titre à utiliser avec un groupe donné de commandes utilisé lors du chargement d'un système d'exploitation.

Le symbole dièse (#) permet d'insérer des commentaires dans le fichier de configuration du menu.

2.7.2. Structure des fichiers de configuration

Le fichier de configuration de l'interface menu de GRUB est `/boot/grub/grub.conf`. Les commandes servant à la définition des préférences générales pour l'interface menu sont placées dans le haut du fichier, suivies des différentes entrées relatives à chacun des systèmes d'exploitation ou noyaux énumérés dans le menu.

L'extrait ci-dessous correspond à un fichier de configuration du menu de GRUB très simple servant au démarrage de Red Hat Linux ou de Microsoft Windows 2000:

```
default=0
timeout=10
splashimage=(hd0,0)/grub/splash.xpm.gz

# section to load linux
title Red Hat Linux (2.4.18-5.47)
        root (hd0,0)
        kernel /vmlinuz-2.4.18-5.47 ro root=/dev/sda2
        initrd /initrd-2.4.18-5.47.img

# section to load Windows 2000
title windows
        rootnoverify (hd0,0)
        chainloader +1
```

Ce fichier invite GRUB à construire un menu avec Red Hat Linux comme système d'exploitation par défaut, réglé pour un démarrage automatique après 10 secondes. Deux sections sont disponibles, une

pour chaque système d'exploitation avec les commandes spécifiques de la table de partition de chaque système.



Remarque

Notez bien que le paramètre par défaut est spécifié sous la forme d'un chiffre. Ceci se rapporte à la première ligne `title` que GRUB rencontre. Si vous voulez que `windows` soit le paramètre par défaut, changez la valeur `default=0` en `default=1`.

Le paramétrage d'un fichier de menu de configuration GRUB pour le démarrage multiple de systèmes d'exploitation va au-delà de la portée de ce chapitre. Ainsi, pour obtenir une liste des ressources supplémentaires, reportez-vous à la Section 2.11.

2.8. LILO

LILO, un acronyme désignant *LI*nux *LO*ader, est utilisé depuis de nombreuses années pour démarrer Linux sur les systèmes x86. Même si GRUB est à présent le chargeur de démarrage par défaut, certains utilisateurs préfèrent utiliser LILO parce qu'ils le connaissent mieux alors que d'autres doivent le choisir par nécessité, GRUB pouvant en effet rencontrer des problèmes lors de l'amorçage de certains matériels.

2.8.1. LILO et le processus de démarrage x86

Cette section traite de façon plus détaillée le rôle spécifique joué par LILO lors du démarrage d'un système x86. Pour une présentation détaillée du processus de démarrage global, voir la Section 1.2.

Le chargement en mémoire de LILO est quasiment identique à celui de GRUB, à la différence près qu'il s'agit d'un chargeur deux étapes uniquement.

1. *Le chargeur Étape 1 ou primaire est lu en mémoire par le BIOS à partir du MBR*[2]. Le chargeur de démarrage primaire existe sur moins de 512 octets d'espace disque dans le MBR. Sa seule tâche consiste à charger le chargeur de démarrage Étape 2 et à lui transférer les informations concernant la géométrie du disque.
2. *Le chargeur de démarrage Étape 2 ou secondaire est lu en mémoire.* Le chargeur de démarrage secondaire affiche l'écran initial Red Hat Linux. Cet écran vous permet de sélectionner le système d'exploitation ou noyau Linux à démarrer.
3. *Le chargeur de démarrage Étape 2 lit en mémoire le système d'exploitation ou noyau et `initrd`.* Une fois que LILO détermine le système d'exploitation à démarrer, il le charge en mémoire et cède le contrôle de la machine à ce système d'exploitation.

Une fois que le chargeur de démarrage Étape 2 est en mémoire, LILO affiche l'écran Red Hat Linux initial avec les différents systèmes d'exploitation ou noyaux qu'il doit démarrer (selon sa configuration). Si, par défaut, vous n'avez installé que Red Hat Linux et n'avez rien changé dans le fichier de configuration LILO, **linux** sera votre seule option. Si en revanche, le système dispose de multiples processeurs, plusieurs options seront disponibles: une option **linux-up** pour un noyau à processeur simple et **linux** pour le noyau à processeur multiple (SMP). Si vous avez configuré LILO pour qu'il démarre également d'autres systèmes d'exploitation, ces sélections apparaîtront sur cet écran.

Les touches fléchées permettent à l'utilisateur de mettre en surbrillance le système d'exploitation retenu et la touche [Entrée] amorce le processus de démarrage.

2. Pour en savoir plus sur le BIOS et le MBR, voir la Section 1.2.1.

Pour accéder à une invite `boot:` appuyez sur [Ctrl]-[X].

2.8.2. LILO contre GRUB

En général, LILO fonctionne d'une façon similaire à GRUB, mais il existe cependant trois différences importantes:

- Il ne dispose pas d'une interface de commande interactive.
- Il stocke les informations sur l'emplacement du noyau ou du système d'exploitation qu'il doit charger sur le MBR.
- Il ne peut pas lire les partitions ext2.

La première différence est que l'invite de commande LILO n'est pas interactive et n'autorise qu'une commande avec des arguments.

Les deux autres différences sont les suivantes: si vous modifiez le fichier de configuration LILO ou installez un nouveau noyau, vous devez réécrire le chargeur de démarrage Étape 1 sur le MBR en exécutant la commande suivante:

```
/sbin/lilo -v -v
```

Cela est beaucoup plus risqué que la méthode de GRUB car un bloc de démarrage maître mal configuré empêche tout simplement le démarrage du système. Avec GRUB, si le fichier de configuration est mal configuré, le programme va tout simplement revenir par défaut à son interface de ligne de commande, à partir de laquelle l'utilisateur peut démarrer manuellement le système.



Astuce

Si vous mettez à niveau le noyau à l'aide de l'application **Agent de mise à jour Red Hat**, le MBR sera mis à jour automatiquement. Pour plus d'informations sur RHN, reportez-vous à l'URL suivante: https://rhn.redhat.com

2.9. Options dans `/etc/lilo.conf`

Le fichier de configuration de LILO est `/etc/lilo.conf`. Les commandes `/sbin/lilo` utilisent ce fichier afin de déterminer ce qui devra être écrit sur le MBR.



Avertissement

Avant d'apporter toute modification au fichier `/etc/lilo.conf`, assurez-vous de bien faire une copie de sauvegarde du fichier. Assurez-vous également que vous disposez d'une disquette de démarrage afin de pouvoir modifier le MBR en cas de problème. Pour plus d'informations sur la création d'une disquette de démarrage, consultez les pages de manuel relatives à `mkbootdisk`.

Le fichier `/etc/lilo.conf` est utilisé par la commande `/sbin/lilo` pour préciser le système d'exploitation ou le noyau à démarrer, ainsi que l'emplacement de son installation.

Un exemple de fichier `/etc/lilo.conf` ressemble à l'extrait suivant:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
message=/boot/message
lba32
default=linux

image=/boot/vmlinuz-2.4.0-0.43.6
 label=linux
 initrd=/boot/initrd-2.4.0-0.43.6.img
 read-only
 root=/dev/hda5

other=/dev/hda1
 label=dos
```

Cet exemple illustre un système configuré pour démarrer deux systèmes d'exploitation: Red Hat Linux et DOS. Ci-après figure un examen plus détaillé de ces lignes:

- `boot=/dev/hda` — indique à LILO de s'installer sur le premier disque dur du premier contrôleur IDE.
- `map=/boot/map` — localise le fichier map. Pour une utilisation normale, cette ligne ne doit pas être modifiée.
- `install=/boot/boot.b` — indique à LILO d'installer le fichier spécifié comme nouveau secteur de démarrage. Pour une utilisation normale, cette ligne ne doit pas être modifiée. Si la ligne `install` est absente, LILO désignera `/boot/boot.b` comme fichier à utiliser par défaut.
- `prompt` — indique à LILO de vous montrer ce qui est référencé à la ligne `message`. Bien qu'il ne soit pas recommandé de supprimer la ligne `prompt`, si vous le faites, vous pouvez toujours obtenir une invite en appuyant longuement [Maj] pendant que le démarrage de votre machine commence.
- `timeout=50` — établit la durée pendant laquelle LILO attendra une saisie de l'utilisateur avant de passer au démarrage de l'entrée spécifiée à la ligne `default`. Cette durée est mesurée en dixièmes de secondes, 50 étant la valeur par défaut.
- `message=/boot/message` — renvoie à l'écran que LILO affiche pour vous permettre de sélectionner le système d'exploitation ou noyau à démarrer.
- `lba32` — décrit la géométrie du disque dur à LILO. L'entrée `linear` est également courante. Nous vous conseillons de ne pas modifier cette ligne, à moins que vous ne soyez vraiment certain des conséquences. Dans le cas contraire, vous pourriez placer votre système dans une situation où le démarrage sera impossible.
- `default=linux` — se rapporte au système d'exploitation que LILO doit charger par défaut à partir des options listées sous cette ligne. Le nom `linux` renvoie à la ligne `label` en dessous dans chacune des options de démarrage.
- `image=/boot/vmlinuz-2.4.0-0.43.6` — spécifie le noyau Linux à démarrer avec cette option de démarrage particulière.
- `label=linux` — précise l'option de système d'exploitation à l'écran LILO. Dans ce cas, il s'agit également du nom auquel la ligne `default` fait référence.
- `initrd=/boot/initrd-2.4.0-0.43.6.img` — se rapporte à l'image du *disque ram initial* utilisée au démarrage pour initialiser et démarrer les dispositifs permettant l'amorçage du noyau. Le disque ram initial est un ensemble de pilotes spécifiques nécessaires à l'opération d'une carte SCSI, d'un disque dur ou de tout autre dispositif entrant dans le chargement du noyau. Ne partagez jamais des disques ram initiaux entre plusieurs machines.

- `read-only` — précise que la partition root (voir la ligne `root` ci-dessous) est en lecture-seule et ne peut pas être modifiée lors du processus de démarrage.
- `root=/dev/hda5` — indique à LILO quelle partition de disque utiliser comme partition root.
- `other=/dev/hda1` — indique la partition contenant DOS.

2.10. Changement de niveau d'exécution au démarrage

Sous Red Hat Linux, il est possible de changer le niveau d'exécution par défaut au démarrage.

Si vous utilisez LILO comme chargeur de démarrage, accédez à l'invite `boot:` en tapant [Ctrl]-[X]. Ensuite, entrez:

`linux` **<*numéro-d'exécution*>**

Dans cette commande, remplacez <*numéro-d'exécution*> par le numéro du niveau d'exécution auquel vous souhaitez que le démarrage soit amorcé (de **1** à **5**), ou les mots **single** ou **emergency**.

Si vous utilisez GRUB comme chargeur de démarrage, suivez les étapes suivantes:

- À l'écran de chargeur de démarrage graphique GRUB, sélectionnez l'étiquette (ou label) de démarrage **Red Hat Linux** et appuyez sur [e] pour l'éditer.
- À l'aide de la flèche bas, allez jusqu'à la ligne de noyau et appuyez sur [e] pour l'éditer.
- À l'invite, tapez le numéro du niveau d'exécution souhaité (de **1** à **5**) ou les mots **single** ou **emergency** et appuyez sur [Entrée].
- Vous retournerez à l'écran GRUB avec les informations sur le noyau. Appuyez sur la touche [b] pour démarrer le système.

Pour plus d'informations sur les niveaux d'exécution, voir la Section 1.4.1.

2.11. Ressources supplémentaires

Ce chapitre se limite à une introduction à GRUB et LILO. Consultez les ressources suivantes si vous souhaitez en savoir plus sur le fonctionnement de GRUB et LILO.

2.11.1. Documentation installée

- `/usr/share/doc/grub-`<*numéro-de-version*> — Ce répertoire contient un certain nombre d'informations sur l'utilisation et la configuration de GRUB. Le <*numéro-de-version*> dans le chemin d'accès vers ce fichier correspond à la version du paquetage GRUB installé.
- La page d'info de GRUB, accessible en tapant la commande `info grub`, contient des leçons, ainsi qu'un manuel de référence pour les utilisateurs et les programmeurs et un Forum Aux Questions (FAQ).
- `/usr/share/doc/lilo-`<*numéro-de-version*>`/` — Ce répertoire contient un nombre important d'informations sur l'utilisation et la configuration de LILO. Plus précisément, le sous-répertoire `doc/` contient un fichier postscript appelé `User_Guide.ps` qui contient des informations très utiles. Le <*numéro-de-version*> dans le chemin d'accès vers ce répertoire correspond à la version du paquetage GRUB installé.

2.11.2. Sites Web utiles

- http://www.gnu.org/software/grub/ — La page d'accueil du projet GRUB de GNU. Ce site contient des informations concernant l'état du développement de GRUB ainsi qu'un FAQ.
- http://www.uruk.org/orig-grub/ — La documentation originale de GRUB, telle qu'elle existait avant que le projet ne soit passé à la Free Software Foundation pour un plus développement poussé.
- http://www.redhat.com/mirrors/LDP/HOWTO/mini/Multiboot-with-GRUB.html — examine les différents usages possibles de GRUB, y compris le démarrage de systèmes d'exploitation autres que Linux.
- http://www.linuxgazette.com/issue64/kohli.html — Un article d'introduction traitant de la configuration de GRUB sur un système, à partir des toutes premières étapes. Il inclut entre autres un aperçu des options de la ligne de commande de GRUB.
- http://www.tldp.org/HOWTO/mini/LILO.html — Ce mini-HOWTO examine différentes utilisations de LILO, y compris le démarrage de systèmes d'exploitation autres que Linux.



Chapitre 3.
Structure d’un système de fichiers

3.1. Pourquoi partager une structure commune?

La structure du système de fichiers d’un système d’exploitation est son niveau d’organisation le plus bas. Presque toutes les façons dont un système d’exploitation interagit avec ses utilisateurs, ses applications et son modèle de sécurité dépendent de la façon dont il stocke ses fichiers dans un périphérique de stockage de base (généralement une unité de disque dur). Il est impératif, et ce pour nombre de raisons, que les utilisateurs, ainsi que les programmes puissent compter sur une ligne directrice commune afin de savoir où lire et écrire des fichiers.

Les systèmes de fichiers peuvent être définis selon deux types différents de catégories logiques de fichiers :

- Fichiers partageables/fichiers non-partageables
- Fichiers variables/fichiers statiques

Les fichiers *partageables* sont accessibles à partir de différents hôtes, alors que les fichiers *non-partageables* ne sont pas disponibles aux autres hôtes. Les fichiers *variables* peuvent être modifiés à tout moment, sans aucune intervention. Les fichiers *statiques*, tels que la documentation ou les fichiers binaires, ne peuvent être changés sans l’action directe de l’administrateur système ou d’un agent mis en place par ce dernier afin d’accomplir cette tâche.

Nous définissons ces fichiers de cette manière en raison des différents types d’autorisations données aux répertoires qui les contiennent. La façon dont le système d’exploitation et ses utilisateurs utilisent les fichiers détermine le répertoire où ces fichiers doivent être placés, selon qu’il est monté pour la lecture-seule ou pour la modification, ainsi que le niveau d’accès permis pour chaque fichier. Le niveau le plus élevé de cette organisation est crucial car l’accès aux sous-répertoires sous-jacents pourrait être limité ou des problèmes de sécurité pourraient survenir si le niveau le plus élevé est mal organisé ou s’il ne dispose pas d’une structure largement utilisée.

Toutefois, le fait d’avoir une structure ne signifie pas grand chose à moins qu’elle ne soit un standard. En effet, des structures concurrentes peuvent créer plus de problèmes qu’elles n’en règlent. Pour cette raison, Red Hat a choisi la structure de système de fichiers la plus utilisée et l’a étendue légèrement pour la prise en charge de fichiers spéciaux spécifiques à Red Hat Linux.

3.2. Aperçu du FHS (’Filesystem Hierarchy Standard’)

Red Hat adhère au *FHS* (de l’anglais ’*Filesystem Hierarchy Standard*’), un document de collaboration définissant les noms et les emplacements de nombreux fichiers et répertoires. Nous continuerons à respecter cette norme pour garantir la conformité de Red Hat Linux avec le FHS.

Le document FHS actuel est la référence faisant autorité pour tout système de fichiers compatible avec le standard FHS, mais le standard comprend de nombreuses zones indéfinies ou extensibles. Cette section donne un aperçu de la norme et une description des éléments du système de fichiers qui ne sont pas couverts par celle-ci.

La norme complète peut être consultée à l’adresse suivante:

http://www.pathname.com/fhs

La conformité avec la norme signifie beaucoup, mais les deux aspects les plus importants sont la compatibilité avec d’autres systèmes également conformes et la possibilité de monter la partition

/usr en lecture-seule car elle contient des fichiers exécutables courants et n'a pas pour vocation d'être modifiée par les utilisateurs. Du fait que le répertoire /usr peut être monté en lecture-seule, il est possible de monter /usr depuis le CD-ROM ou un autre ordinateur par le biais d'un NFS en lecture-seule.

3.2.1. Organisation de FHS

Les répertoires et les fichiers mentionnés ici sont un petit sous-ensemble de ceux qui sont spécifiés par le document FHS. Consultez le document FHS le plus récent pour obtenir des renseignements complets.

3.2.1.1. Le répertoire /dev

Le répertoire /dev contient des entrées de système de fichiers représentant des périphériques connectés au système. Ces fichiers sont essentiels au bon fonctionnement du système.

3.2.1.2. Le répertoire /etc

Le répertoire /etc est réservé aux fichiers de configuration locaux sur votre ordinateur. Tous les fichiers binaires qui se trouvaient auparavant dans /etc devraient dorénavant aller dans /sbin ou, si possible, dans /bin.

Les répertoires X11 et skel doivent être des sous-répertoires du répertoire /etc:

```
/etc
  |- X11
  |- skel
```

Le répertoire X11 est destiné aux fichiers de configuration X11, tels que XF86Config. Le répertoire skel est consacré aux fichiers utilisateur "squelette", utilisés pour remplir un répertoire personnel lors de la création d'un nouvel utilisateur.

3.2.1.3. Le répertoire /lib

Le répertoire /lib ne devrait contenir que les bibliothèques nécessaires à l'exécution de fichiers binaires dans /bin et /sbin. Ces images de bibliothèques partagées sont particulièrement importantes pour le démarrage du système et l'exécution de commandes dans le système de fichiers racine.

3.2.1.4. Le répertoire /mnt

Le répertoire /mnt se réfère aux systèmes de fichiers montés de façon temporaire, tels que les CD-ROM et les disquettes.

3.2.1.5. Le répertoire /opt

Le répertoire /opt fournit un endroit pour stocker des paquetages de logiciels d'applications statiques de grande taille.

Lorsque l'on veut éviter de mettre les fichiers d'un paquetage donné dans le système de fichiers, /opt fournit un système organisationnel logique et prévisible sous le répertoire du paquetage en question. Cela donne à l'administrateur système une façon facile de déterminer le rôle de chaque fichier d'un paquetage donné.

Par exemple, si `sample` est le nom d'un paquetage logiciel situé dans `/opt`, alors tous ses fichiers pourraient être placés dans des répertoires à l'intérieur de `/opt/sample`, tels que `/opt/sample/bin` pour les fichiers binaires et `/opt/sample/man` pour les pages de manuel.

Les paquetages de grande taille qui contiennent de nombreux sous-paquetages différents exécutant chacun une tâche spécifique, vont également dans le répertoire `/opt`, leur donnant ainsi une façon standard de s'organiser. Pour reprendre notre exemple, le paquetage `sample` pourrait contenir différents outils allant chacun dans un sous-répertoire qui lui est propre, tel que `/opt/sample/tool1` et `/opt/sample/tool2`, qui à son tour peut avoir ses propres répertoires `bin`, `man` ou autres répertoires semblables.

3.2.1.6. Le répertoire `/proc`

Le répertoire `/proc` contient des fichiers spéciaux qui extraient des informations à partir du ou envoient des informations au noyau.

Étant donné l'immense variété de données disponibles dans `/proc` et les différentes façons dont ce répertoire peut être utilisé pour communiquer avec le noyau, un chapitre entier a été consacré à ce sujet. Pour plus d'informations, consultez le Chapitre 5.

3.2.1.7. Le répertoire `/sbin`

Le répertoire `/sbin` est conçu pour les fichiers exécutables qui ne sont utilisés que par les utilisateurs racine. Les fichiers exécutables dans `/sbin` ne sont utilisés que pour démarrer et monter `/usr` et exécuter des opérations de remise en état du système. FHS indique ce qui suit:

> "`/sbin` contient généralement des fichiers essentiels pour le démarrage du système, en plus des fichiers binaires figurant dans `/bin`. Tout ce qui est exécuté après `/usr` est supposé monté (lorsqu'il n'y a pas de problème) et doit être placé dans `/usr/sbin`. Les fichiers binaires d'administration du système exclusivement locaux doivent être placés dans le répertoire `/usr/local/sbin`."

Au minimum, les programmes suivants doivent être présents dans `/sbin`:

```
arp, clock,
getty, halt,
init, fdisk,
fsck.*, grub,
ifconfig, lilo,
mkfs.*, mkswap,
reboot, route,
shutdown, swapoff,
swapon, update
```

3.2.1.8. Le répertoire `/usr`

Le répertoire `/usr` est destiné aux fichiers pouvant être partagés sur l'ensemble d'un site. Le répertoire `/usr` a généralement sa propre partition et devrait être montable en lecture-seule. Les répertoires suivants doivent être des sous-répertoires de `/usr`:

```
/usr
  |- bin
  |- dict
  |- doc
  |- etc
  |- games
```

```
  |- include
  |- kerberos
  |- lib
  |- libexec
  |- local
  |- sbin
  |- share
  |- src
  |- tmp -> ../var/tmp
  |- X11R6
```

Le répertoire `bin` contient des fichiers exécutables, `doc` contient des pages de documentation, `etc` contient des fichiers de configuration pour l'ensemble du système, `games` est pour les jeux, `include` contient des fichiers d'en-tête C, `kerberos` contient des fichiers binaires et d'autres éléments pour Kerberos et, enfin, `lib` contient des fichiers objet et des bibliothèques qui ne sont pas destinés à être utilisés directement par les utilisateurs ou les scripts shell. Le répertoire `libexec` contient de petits programmes d'aide appelés par d'autres programmes, `sbin` est pour les fichiers binaires d'administration du système (ceux qui n'appartiennent pas à `/sbin`), `share` contient des fichiers qui ne sont pas spécifiques à l'architecture, `src` est pour le code source et `X11R6` est pour le système X Window (**XFree86** sur Red Hat Linux).

3.2.1.9. Le répertoire `/usr/local`

FHS indique ce qui suit:

> "La hiérarchie `/usr/local` est destinée à être installée par l'administrateur système lors de l'installation locale du logiciel. Elle doit être à l'abri de toute réécriture lors de la mise à jour du logiciel système. Elle peut être utilisée pour des programmes et des données partageables entre un groupe d'ordinateurs, mais ne figurant pas dans `/usr`."

Le répertoire `/usr/local` est semblable, de par sa structure, au répertoire `/usr`. Il contient les sous-répertoires suivants, qui sont semblables, de par leur fonction, à ceux qui se trouvent dans le répertoire `/usr`:

```
/usr/local
       |- bin
       |- doc
       |- etc
       |- games
       |- include
       |- lib
       |- libexec
       |- sbin
       |- share
       |- src
```

3.2.1.10. Le répertoire `/var`

Comme FHS exige que vous soyez en mesure de monter `/usr` en lecture-seule, tous les programmes qui écrivent des fichiers journaux ou ont besoin de répertoires `spool` ou `lock` devraient probablement les écrire dans le répertoire `/var`. FHS indique que `/var` est pour :

> "...les fichiers de données variables. Ceci comprend les répertoires et fichiers spool, les données administratives et de journalisation, de même que les fichiers transitoires et temporaires."

Les répertoires suivants peuvent être des sous-répertoires de `/var` :

```
/var
  |- account
  |- arpwatch
  |- cache
  |- crash
  |- db
  |- empty
  |- ftp
  |- gdm
  |- kerberos
  |- lib
  |- local
  |- lock
  |- log
  |- mail -> spool/mail
  |- mailman
  |- named
  |- nis
  |- opt
  |- preserve
  |- run
  +- spool
       |- anacron
       |- at
       |- cron
       |- fax
       |- lpd
       |- mail
       |- mqueue
       |- news
       |- rwho
       |- samba
       |- slrnpull
       |- squid
       |- up2date
       |- uucp
       |- uucppublic
       |- vbox
       |- voice
  |- tmp
  |- tux
  |- www
  |- yp
```

Les fichiers journaux tels que `messages` et `lastlog` vont dans `/var/log`. Le répertoire `/var/lib/rpm` contient aussi les bases de données système RPM. Les fichiers lock vont dans `/var/lock`, généralement dans des répertoires spécifiques aux programmes qui utilisent ces fichiers. Le répertoire `/var/spool` comprend des sous-répertoires pour divers systèmes ayant besoin de stocker des fichiers de données.

3.2.2. `/usr/local` in Red Hat Linux

Dans Red Hat Linux, l'utilisation prévue pour `/usr/local` est légèrement différente de celle qui est spécifiée par FHS. FHS indique que `/usr/local` devrait se trouver là où vous stockez des logiciels devant rester à l'abri des mises à jour du logiciel système. Du fait que les mises à jour du système à partir de Red Hat s'effectuent en toute sécurité à l'aide du système `rpm` et de **Gnome-RPM**, il ne vous est pas nécessaire de protéger des fichiers en les plaçant dans `/usr/local`. Il vous est plutôt recommandé d'utiliser `/usr/local` pour y placer les logiciels locaux de votre ordinateur.

Par exemple, imaginons que vous ayez monté `/usr` par le biais d'un NFS en lecture-seule à partir d'un hôte appelé `jake`. Si vous désirez installer un paquetage ou un programme, mais que vous n'avez pas l'autorisation d'apporter des modifications dans `jake`, vous devriez alors l'installer sous `/usr/local`. Par la suite peut-être, si vous réussissez à convaincre l'administrateur système de `jake` d'installer le programme dans `/usr`, vous pourrez le désinstaller du répertoire `/usr/local`.

3.3. Emplacement de fichiers spéciaux

Red Hat étend légèrement la structure FHS pour prendre en charge les fichiers spéciaux utilisés par Red Hat Linux.

La plupart des fichiers appartenant à *'Red Hat Package Manager'* (ou *RPM*) se trouvent dans le répertoire `/var/lib/rpm/`. Pour avoir plus de détails sur RPM, reportez-vous au chapitre intitulé *Gestion de paquetage avec RPM* du *Guide de personnalisation de Red Hat Linux*.

Le répertoire `/var/spool/up2date/` contient des fichiers utilisés par l'**Agent de mise à jour Red Hat**, y compris des informations de titres RPM. Cet emplacement peut aussi être utilisé pour stocker temporairement des RPM téléchargés lorsque vous mettez à jour votre système. Pour plus d'informations sur le Réseau Red Hat, voyez le site Web à l'adresse suivante https://rhn.redhat.com/.

Un autre emplacement spécifique à Red Hat Linux est le répertoire `/etc/sysconfig/` directory. ce répertoire stocke un grand nombre d'informations de configuration. De nombreux scripts lancés au démarrage utilisent les fichiers de ce répertoire. Consultez le Chapitre 4 pour obtenir plus d'informations sur le contenu de ce répertoire et le rôle de ces fichiers dans le processus de démarrage.

Enfin, un dernier répertoire à connaître est le répertoire `/initrd/`. Il est vide, mais est utilisé comme point de montage critique pendant le processus de démarrage.


Avertissement

Ne supprimez le répertoire `/initrd/` sous aucun prétexte. L'enlever empêcherait votre système de démarrer, avec un message d'erreur panique du noyau.



Chapitre 4. Le répertoire sysconfig

Le répertoire /etc/sysconfig/ est l'endroit où sont stockés de nombreux fichiers de configuration de Red Hat Linux.

Ce chapitre souligne certains des fichiers situés dans le répertoire /etc/sysconfig/, leur fonction et leur contenu. Ces informations ne prétendent pas être exhaustives car nombre de ces fichiers sont une série d'options qui ne sont utilisées que dans de circonstances spécifiques et plutôt rares.

4.1. Fichiers contenus dans le répertoire /etc/sysconfig/

Les fichiers suivants se trouvent généralement dans le répertoire /etc/sysconfig/:

- amd
- apmd
- arpwatch
- authconfig
- cipe
- clock
- desktop
- dhcpd
- firstboot
- gpm
- harddisks
- hwconf
- i18n
- identd
- init
- ipchains
- iptables
- irda
- keyboard
- kudzu
- mouse
- named
- netdump
- network
- ntpd
- pcmcia
- radvd

- rawdevices
- redhat-config-securitylevel
- redhat-config-users
- redhat-logviewer
- samba
- sendmail
- soundcard
- spamassassin
- squid
- tux
- ups
- vncservers
- xinetd



Remarque

Si certains des fichiers énumérés ci-dessus ne sont pas présents dans le répertoire /etc/sysconfig/, le programme auquel ils associés ne pourra pas être installé.

4.1.1. /etc/sysconfig/amd

Le fichier /etc/sysconfig/amd contient différents paramètres utilisés par amd pour permettre le montage et le démontage automatique de systèmes de fichiers.

4.1.2. /etc/sysconfig/apmd

Le fichier /etc/sysconfig/apmd est utilisé par apmd en tant que configuration pour indiquer ce qu'il faut démarrer/arrêter/modifier en cas de suspension ou de reprise des opérations. Il est configuré pour activer ou désactiver apmd pendant le démarrage, en fonction de la prise en charge ou non de la technologie '*Advanced Power Management*' (ou *APM*) par votre matériel d'une part ou de votre décision de ne pas l'utiliser d'autre part. Le démon de contrôle apm fonctionne avec le code de gestion d'énergie au sein du noyau Linux. Il permet notamment d'avertir les utilisateurs d'ordinateurs portables lorsque le niveau de la batterie est bas ou lorsqu'il a un problème avec des paramètres ayant un lien avec une source électrique.

4.1.3. /etc/sysconfig/arpwatch

Le fichier /etc/sysconfig/arpwatch est utilisé pour transmettre des arguments au démon arpwatch lors du démarrage. Le démon arpwatch maintient une table d'adresses Ethernet MAC et leurs parités d'adresses IP. Pour de plus amples informations sur les paramètres que vous pouvez utiliser dans ce fichier, tapez man arpwatch. Par défaut, ce fichier règle le propriétaire du processus arpwatch sur l'utilisateur pcap.

4.1.4. /etc/sysconfig/authconfig

Le fichier /etc/sysconfig/authconfig détermine le type d'autorisation à utiliser sur l'ordinateur hôte. Il contient une ou plusieurs des lignes suivantes:

- USEMD5=<*valeur*>, où <*valeur*> correspond à un des éléments ci-dessous:
  - yes — MD5 est utilisé pour l'authentification.
  - no — MD5 n'est pas utilisé pour l'authentification.

- USEKERBEROS=<*valeur*>, où <*valeur*> correspond à un des éléments ci-dessous:
  - yes — Kerberos est utilisé pour l'authentification.
  - no — Kerberos n'est pas utilisé pour l'authentification.

- USELDAPAUTH=<*valeur*>, où <*valeur*> correspond à un des éléments ci-dessous:
  - yes — LDAP est utilisé pour l'authentification.
  - no — LDAP n'est pas utilisé pour l'authentification.

4.1.5. /etc/sysconfig/clock

Le fichier /etc/sysconfig/clock contrôle l'interprétation des valeurs lues à partir de l'horloge matérielle du système.

Les valeurs correctes sont les suivantes:

- UTC=<*valeur*>, où <*valeur*> correspond à l'une des valeurs booléennes suivantes:
  - true ou yes — indique que l'horloge matérielle est réglée sur l'heure universelle (celle du méridien de Greenwich).
  - false ou no — indique que l'horloge matérielle est réglée sur l'heure locale.

- ARC=<*valeur*>, où <*valeur*> correspond à:
  - true ou yes — indique que le décalage de 42 ans de la console ARC est activé. Ce paramètre ne s'applique qu'aux systèmes Aplha basés sur ARC ou AlphaBIOS. Toute autre valeur indique que l'époque UNIX normale est la référence.
- SRM=<*valeur*>, où <*valeur*> correspond à:
  - true ou yes — indique que l'époque 1900 de la console SRM est activée. Ce paramètre ne s'applique qu'aux systèmes Alpha basés sur SRM. Toute autre valeur indique que l'époque UNIX normale est la référence.
- ZONE=<*nom-de-fichier*> — indique le fichier de fuseau horaire dans /usr/share/zoneinfo dont /etc/localtime est une copie, comme par exemple:
  ZONE="America/New York"

Des versions précédentes de Red Hat Linux utilisaient les valeurs suivantes (qui ne sont désormais plus valables):

- CLOCKMODE=<*valeur*>, où <*valeur*> correspond à l'une des valeurs suivantes:

- GMT — indique que l'horloge est réglée sur l'heure universelle (UTC: 'Universal Time Clock' ou GMT: 'Greenwich Mean Time').
- ARC — indique que le décalage de 42 ans de la console ARC est activé. (pour les systèmes basés sur Alpha seulement).

4.1.6. /etc/sysconfig/desktop

Le fichier /etc/sysconfig/desktop spécifie le gestionnaire de bureau devant être exécuté, comme par exemple:

```
DESKTOP="GNOME"
```

4.1.7. /etc/sysconfig/dhcpd

Le fichier /etc/sysconfig/dhcpd est utilisé pour transmettre des arguments au démon dhcpd lors du démarrage. Le démon dhcpd met en oeuvre les protocoles 'Dynamic Host Configuration Protocol' (ou DHCP) et Internet Bootstrap Protocol (ou BOOTP). DHCP et BOOTP assignent des noms d'hôtes aux ordinateurs sur le réseau. Pour de plus amples informations sur les paramètres pouvant être utilisés dans ce fichier, consultez la page de manuel relative à dhcpd.

4.1.8. /etc/sysconfig/firstboot

Depuis Red Hat Linux 8.0, lors du premier démarrage du système, le programme /sbin/init appelle le script etc/rc.d/init.d/firstboot, qui à son tour lance l'**Agent de paramétrage**. Cette application permet à l'utilisateur d'installer les dernières mises à jour ainsi que les applications et la documentation supplémentaires.

Le fichier /etc/sysconfig/firstboot indique à l'application **Agent de paramétrage** de ne pas s'exécuter lors de prochains démarrages. Pour la lancer lors du prochain démarrage du système, supprimez /etc/sysconfig/firstboot et exécutez chkconfig --level 5 firstboot on.

4.1.9. /etc/sysconfig/gpm

Le fichier /etc/sysconfig/gpm est utilisé pour transmettre des arguments au démon gpm lors du démarrage. Le démon gpm permet l'accélération de la souris et le collage à en cliquant sur le bouton central de la souris. Pour de plus amples informations sur les paramètres pouvant être utilisés dans ce fichier, consultez la page de manuel relative à gpm. Par défaut, il règle le périphérique souris sur /dev/mouse.

4.1.10. /etc/sysconfig/harddisks

Le fichier /etc/sysconfig/harddisks vous permet de régler votre ou vos disque(s) dur(s). L'administrateur peut également utiliser /etc/sysconfig/hardiskhd[a-h], pour configurer les paramètres de disques durs spécifiques.



Avertissement

Réfléchissez bien avant d'apporter toute modification à ce fichier. Si vous changez les valeurs par défaut contenues dans ce fichier, vous risquez de corrompre toutes les données de votre ou vos disque(s) dur(s).

Le fichier `/etc/sysconfig/harddisks` peut contenir les éléments suivants:

- `USE_DMA=1`, où la valeur 1 active DMA. Néanmoins, avec certaines combinaisons jeux de puces/disque dur, cette DMA peut entraîner une corruption des données. *Avant de l'activer, vérifiez bien la documentation de votre disque dur ou demandez conseil au fabricant.*
- `Multiple_IO=16`, où la valeur 16 autorise plusieurs secteurs par interruption d'entrée/sortie. Lorsqu'elle est activée, cette fonction réduit le temps de gestion du système de 30 à 50%. *Utilisez cette fonction avec prudence.*
- `EIDE_32BIT=3` active le support E/S (E)IDE 32-bit par une carte d'interface.
- `LOOKAHEAD=1` active l'anticipation en lecture du disque.
- `EXTRA_PARAMS=` précise l'endroit où peuvent être ajoutés des paramètres supplémentaires.

4.1.11. /etc/sysconfig/hwconf

Le fichier `/etc/sysconfig/hwconf` affiche la liste de tout le matériel que `kudzu` a détecté sur votre ordinateur, ainsi que des informations sur les pilotes utilisés, l'ID du fabricant et du périphérique. Le programme `kudzu` détecte et configure le matériel nouveau et/ou changé sur un système. Le fichier `/etc/sysconfig/hwconf` n'est pas supposé être modifié manuellement. Dans le cas où vous le feriez, certains périphériques pourraient soudainement apparaître comme étant ajoutés ou supprimés.

4.1.12. /etc/sysconfig/i18n

Le fichier `/etc/sysconfig/i18n` règle la langue par défaut, toute langue prise en charge et la police de caractères par défaut. Par exemple:

```
LANG="en_US.UTF-8"
SUPPORTED="en_US.UTF-8:en_US:en"
SYSFONT="latarcyrheb-sun16"
```

4.1.13. /etc/sysconfig/identd

Le fichier `/etc/sysconfig/identd` est utilisé pour transmettre des arguments au démon `identd` lors du démarrage. Le démon `identd` renvoie le nom d'utilisateur des processus avec connexions TCP/IP ouvertes. Certains des services sur le réseau, comme les serveurs FTP et IRC, entraînent des plaintes et des réponses lentes si `identd` n'est pas en cours d'exécution. Mais en général, `identd` n'est pas un service indispensable; ainsi, si la sécurité est critique, nous vous conseillons de ne pas le lancer. Pour de plus amples informations sur les paramètres pouvant être utilisés dans ce fichier, consultez la page de manuel relative à `identd`. Par défaut, ce fichier ne contient aucun paramètre.

4.1.14. /etc/sysconfig/init

Le fichier /etc/sysconfig/init contrôle l'aspect et le fonctionnement du système pendant le processus de démarrage.

Les valeurs suivantes peuvent être utilisées:

- BOOTUP=<*valeur*>, où <*valeur*> correspond à un des éléments suivants:
  - BOOTUP=color indique un affichage couleur standard au démarrage; la réussite ou l'échec du démarrage des périphériques et des services est représenté par des couleurs différentes.
  - BOOTUP=verbose indique un affichage dans l'ancien style (prolixe) qui fournit des informations plus détaillées qu'un simple message de réussite ou d'échec.
  - Tout autre valeur indique un nouvel affichage, mais sans formatage ANSI.

- RES_COL=<*valeur*>, où <*valeur*> correspond au numéro de la colonne de l'écran où commencer les étiquettes d'état. La valeur par défaut est 60.
- MOVE_TO_COL=<*valeur*>, où <*valeur*> déplace le curseur sur la valeur indiquée dans la ligne RES_COL via la commande echo -en.
- SETCOLOR_SUCCESS=<*valeur*>, où <*valeur*> configure la couleur indiquant la réussite via la commande echo -en. Vert est la couleur par défaut.
- SETCOLOR_FAILURE=<*valeur*>, où <*valeur*> configure la couleur indiquant l'échec via la commande echo -en. Rouge est la couleur par défaut.
- SETCOLOR_WARNING=<*valeur*>, où <*valeur*> configure la couleur indiquant un avertissement via la commande echo -en. Jaune est la couleur par défaut.
- SETCOLOR_NORMAL=<*valeur*>, où <*valeur*> configure la couleur sur 'normal' via la commande echo -en.
- LOGLEVEL=<*valeur*>, où <*valeur*> définit le niveau de connexion initial de la console pour le noyau. La valeur par défaut est 3; 8 signifie tout (y compris le débogage); 1 ne signifie rien d'autre que les paniques du noyau. Le démon syslogd écrasera ce paramètre au démarrage.
- PROMPT=<*valeur*>, où <*valeur*> correspond à l'une des valeurs booléennes suivantes:
  - yes — active le contrôle du mode interactif au clavier.
  - no — désactive le contrôle du mode interactif au clavier.

4.1.15. /etc/sysconfig/ipchains

Le fichier /etc/sysconfig/ipchains contient des informations utilisées par le script d'initialisation ipchains lors de l'établissement du service ipchains.

Ce fichier peut être modifié en tapant la commande /sbin/service ipchains save lorsque des règles ipchains valides sont en place. Ne modifiez pas ce fichier manuellement. Il est préférable d'utiliser la commande /sbin/ipchains pour configurer les règles de filtrage des paquets et ensuite enregistrer les règles dans ce fichier à l'aide de la commande /sbin/service ipchains save.

Il n'est pas recommandé d'utiliser ipchains pour établir des règles de pare-feu car cette commande est plus ou moins obsolète est risque de disparaître des versions futures de Red Hat Linux. Si vous avez besoin d'un pare-feu, utilisez plutôt iptables.

4.1.16. /etc/sysconfig/iptables

Tout comme /etc/sysconfig/ipchains, le fichier /etc/sysconfig/iptables stocke des informations utilisées par le noyau pour configurer des services de filtrage au moment du démarrage ou lors de tout démarrage du service.

Il est déconseillé de modifier ce fichier manuellement à moins que vous ne sachiez exactement comment construire des règles iptables. La manière la plus simple d'ajouter des règles consiste à utiliser l'**Outil de configuration du niveau de sécurité** (redhat-config-securitylevel), la commande /usr/sbin/lokkit ou l'application **GNOME Lokkit** pour créer un pare-feu. En utilisant ces applications, ce fichier sera automatiquement modifié à la fin du processus.

Il est possible de créer des règles manuellement à l'aide de /sbin/iptables: tapez ensuite /sbin/service iptables save pour ajouter les règles au fichier /etc/sysconfig/iptables.

Une fois que ce fichier existe, toutes les règles de pare-feu sauvegardées ici seront conservées lors d'un réamorçage du système ou lors de redémarrage d'un service.

Pour de plus amples informations sur iptables, conslutez le Chapitre 16.

4.1.17. /etc/sysconfig/irda

Le fichier /etc/sysconfig/irda contrôle la configuration des périphériques à infrarouge de votre système lors du démarrage.

Les valeurs suivantes peuvent être utilisées:

- IRDA=<*valeur*>, où <*valeur*> correspond à une des valeurs booléennes suivantes:
  - yes — irattach s'exécute et vérifie de façon périodique si certains périphériques essaient de se connecter au port infrarouge, comme par exemple, un autre bloc-notes qui tente d'effectuer une connexion réseau. Pour que des périphériques à infrarouge fonctionnent sur votre système. cette ligne doit prendre la valeur yes.
  - no — irattach ne s'exécutera pas, empêchant ainsi toute communication avec les périphériques à infrarouge.
- DEVICE=<*valeur*>, où <*valeur*> correspond au périphérique (habituellement un port série) qui gère les connexions à infrarouge.
- DONGLE=<*valeur*>, où <*valeur*> spécifie le type de clé électronique utilisée pour les connexions par infrarouge. Ce paramètre existe pour les personnes utilisant une clé électronique série plutôt que de vrais ports infrarouges. Une clé électronique est un dispositif qui est branché à un port série traditionnel pour la communication par infrarouges. Cette ligne est, par défaut, réglée sur l'inactivité car les ordinateurs bloc-notes dotés de vrais ports à infrarouge sont beaucoup plus fréquents que ceux dotés de clés électroniques ajoutées.
- DISCOVERY=<*valeur*>, où <*valeur*> correspond à une des valeurs booléennes suivantes:
  - yes — lance irattach en mode découverte, ce qui signifie qu'il cherche activement d'autres périphériques à infrarouges. Cette fonction doit être activée pour que l'ordinateur puisse chercher de façon active une connexion infrarouge (c'est-à-dire que l'élément ne prend pas l'initiative de la connexion).
  - no — ne lance pas irattach en mode découverte.

4.1.18. /etc/sysconfig/keyboard

Le fichier /etc/sysconfig/keyboard contrôle le comportement du clavier. Il est possible d'utiliser les valeurs suivantes:

- KEYBOARDTYPE=sun|pc, cette valeur n'est utilisée que sur les systèmes SPARCs. La valeur sun indique qu'un clavier Sun est connecté à /dev/kbd et la valeur pc signifie qu'un clavier PS/2 est connecté à un port PS/2.
- KEYTABLE=<*fichier*>, où <*fichier*> représente le nom d'un fichier de clavier.

  Comme, par exemple: KEYTABLE="us". Les fichiers pouvant être utilisés comme fichiers de clavier commencent dans /lib/kbd/keymaps/i386 et se ramifient de là, en différents types de claviers, portant tous l'étiquette <*fichier*>.kmap.gz. Le premier fichier qui se trouve sous /lib/kbd/keymaps/i386 et qui correspond au paramètre KEYTABLE est utilisé.

4.1.19. /etc/sysconfig/kudzu

Le fichier /etc/sysconfig/kuzdu vous permet de spécifier la détection sécuritaire du matériel de votre ordinateur par kudzu au moment du démarrage. Une détection sécuritaire désactive la détection de ports série.

- SAFE=<*valeur*>, où <*valeur*> correspond à une des valeurs suivantes:
  - yes — kuzdu exécute une détection sécuritaire.
  - no — kuzdu exécute une détection normale.

4.1.20. /etc/sysconfig/mouse

Le fichier /etc/sysconfig/mouse est utilisé pour spécifier des informations sur la souris disponible. Les valeurs suivantes peuvent être utilisées:

- FULLNAME=<*valeur*>, où <*valeur*> fait référence au nom complet du type de souris utilisée.
- MOUSETYPE=<*valeur*>, où <*valeur*> correspond à un des éléments suivants:
  - imps2 — une souris générique USB à roue.
  - microsoft — une souris Microsoft™.
  - mouseman — une souris MouseMan™.
  - mousesystems — une souris Mouse Systems™.
  - ps/2 — une souris PS/2.
  - msbm — une souris bus Microsoft™.
  - logibm — une souris bus Logitech™.
  - atibm — une souris bus ATI™.
  - logitech — une souris Logitech™.
  - mmseries — un ancien modèle de souris MouseMan™.
  - mmhittab — une souris mmhittab.
- XEMU3=<*valeur*>, où <*valeur*> correspond à une des valeurs booléennes suivantes:

  - `yes` — la souris n'a que deux boutons, mais trois boutons de souris devraient être simulés.
  - `no` — la souris a déjà trois boutons.

- `XMOUSETYPE=`<*valeur*>, où <*valeur*> fait référence au type de souris utilisé lors de l'exécution de X Window. Les options dans ce cas sont les mêmes que les paramètres `MOUSETYPE` contenus dans ce même fichier.
- `DEVICE=`<*valeur*>, où <*valeur*> indique périphérique de souris.

De plus, `/dev/mouse` est un lien symbolique qui pointe vers le vrai périphérique de souris.

4.1.21. /etc/sysconfig/named

Le fichier `/etc/sysconfig/named` est utilisé pour transmettre des arguments au démon `named` au moment du démarrage. Le démon `named` est un serveur '*Domain Name System*' (*DNS*) qui met en oeuvre le '*Berkeley Internet Name Domain*' (*BIND*) version 9. Ce serveur maintient une table dont les noms d'hôtes sont attachés à des adresses IP sur le réseau.

Actuellement, seules les valeurs suivantes peuvent être utilisées:

- `ROOTDIR="`<*/quelque/part*>`"`, où <*/quelque/part*> fait référence au chemin d'accès du répertoire d'un environnement chroot sous lequel `named` sera exécuté. Cet environnement chroot doit préalablement être configuré. Tapez `info chroot` pour obtenir de plus amples informations sur la manière de procéder.
- `OPTIONS="`<*valeur*>`"`, où <*valeur*> correspond à toute option listée dans la page de manuel relative à `named`, à l'exception de **`-t`**. Au lieu de **`-t`**, utilisez la ligne de commande `ROOTDIR` ci-dessus.

Pour obtenir de plus amples informations sur les différents paramètres pouvant être utilisés dans ce fichier, consultez la page de manuel relative à `named`. Pour des renseignements détaillés sur la façon de configurer un serveur BIND DNS, reportez-vous au Chapitre 12. Par défaut, le fichier ne contient aucun paramètre.

4.1.22. /etc/sysconfig/netdump

Le fichier `/etc/sysconfig/netdump` est le fichier de configuration du service `/etc/init.d/netdump`. Le service `netdump` envoie à la fois des données 'oops' et des surplus de mémoire sur le réseau. En général, `netdump` n'est pas un service nécessaire; ainsi, ne le lancez que si vous en avez absolument besoin. Pour de plus amples informations sur les paramètres que vous pouvez utilisez dans ce fichier, consultez la page de manuel relative à `netdump`.

4.1.23. /etc/sysconfig/network

Le fichier `/etc/sysconfig/network` est utilisé pour spécifier des informations sur la configuration réseau désirée. Les valeurs suivantes peuvent être utilisées:

- `NETWORKING=`<*valeur*>, où <*valeur*> correspond à une des valeurs booléennes suivantes:
  - `yes` — la mise en réseau devrait être configurée.
  - `no` — la mise en réseau ne devrait pas être configurée.

- `HOSTNAME=<valeur>`, où `<valeur>` devrait être le *le nom de domaine complet* (*FQDN* de l'anglais 'Fully Qualified Domain Name'), comme par exemple `hostname.domain.com`, mais vous pouvez tout à fait utiliser le nom d'hôte de votre choix.



Remarque

Pour assurer la compatibilité avec des logiciels plus anciens que certains utilisateurs pourraient installer (comme par exemple, `trn`), le fichier `/etc/HOSTNAME` devrait contenir les mêmes valeurs qu'ici.

- `GATEWAY=<valeur>`, où `<valeur>` est l'adresse IP de la passerelle réseau.
- `GATEWAYDEV=<valeur>`, où `<valeur>` est le périphérique de passerelle, comme par exemple, `eth0`.
- `NISDOMAIN=<valeur>`, où `<valeur>` est le nom de domaine NIS.

4.1.24. /etc/sysconfig/ntpd

Le fichier `/etc/sysconfig/ntpd` est utilisé pour transmettre des arguments au démon `ntpd` au moment du démarrage. Le démon `ntpd` régule et maintient l'horloge du système pour la synchroniser avec un serveur d'heure standard Internet. Il met en oeuvre la version 4 du protocole NTP (de l'anglais 'Network Time Protocol'). Pour de plus amples informations sur les paramètres que vous pouvez utiliser dans ce fichier, consultez la page suivante à l'aide de votre navigateur: `/usr/share/doc/ntp-<version>/ntpd.htm` (où `<version>` correspond au numéro de la version de `ntpd`). Par défaut, ce fichier règle le propriétaire du processus `ntpd` sur l'utilisateur `ntp`.

4.1.25. /etc/sysconfig/pcmcia

Le frichier `/etc/sysconfig/pcmcia` est utilisé pour préciser des informations de configuration de la carte PCMCIA. Il est possible d'utiliser les valeurs suivantes:

- `PCMCIA=<valeur>`, où `<valeur>` correspond à un des éléments suivants:
  - `yes` — le support PCMCIA doit être activé.
  - `no` — le support PCMCIA ne doit pas être activé.

- `PCIC=<valeur>`, où `<valeur>` correspond à un des éléments suivants:
  - `i82365` — l'ordinateur a un jeu de puces de socket PCMCIA de type i82365.
  - `tcic` — l'ordinateur a un jeu de puces de socket PCMCIA de type tcic.

- `PCIC_OPTS=<valeur>`, où `<valeur>` correspond aux paramètres de synchronisation du pilote de support (i82365 ou tcic).
- `CORE_OPTS=<valeur>`, où `<valeur>` correspond à la liste d'options `pcmcia_core`.
- `CARDMGR_OPTS=<valeur>`, où `<valeur>` correspond à la liste d'options pour le `cardmgr` PCMCIA (comme par exemple, `-q` pour le mode silencieux; `-m` pour chercher des modules de noyau chargeables dans le répertoire spécifié, etc.). Lisez la page de manuel relative à `cardmgr` pour de plus amples informations.

4.1.26. /etc/sysconfig/radvd

Le fichier /etc/sysconfig/radvd est utilisé pour transmettre des arguments au démon radvd au moment du démarrage. Le démon radvd surveille les requêtes du routeur et envoie des messages pour le protocole IP version 6. Ce service permet aux hôtes sur un réseau de modifier de façon dynamique leurs routeurs par défaut, sur la base de ces messages routeurs. Pour obtenir de plus amples informations sur les paramètres que vous pouvez utiliser dans ce fichier, consultez la page de manuel relative à radvd. Par défaut, ce fichier règle le propriétaire du processus radvd sur l'utilisateur radvd.

4.1.27. /etc/sysconfig/rawdevices

Le fichier /etc/sysconfig/rawdevices est utilisé pour configurer les liaisons des périphériques bruts ('raw devices'), comme par exemple:

```
/dev/raw/raw1 /dev/sda1
/dev/raw/raw2 8 5
```

4.1.28. /etc/sysconfig/redhat-config-securitylevel

Le fichier /etc/sysconfig/redhat-config-securitylevel contient toutes les options choisies par l'utilisateur lors de la dernière exécution de l'**Outil de configuration du niveau de sécurité** (redhat-config-securitylevel). Il est fortement déconseillé aux utilisateurs de modifier ce fichier manuellement. Pour obtenir de plus amples informations sur l'**Outil de configuration du niveau de sécurité**, consultez le chapitre intitulé *Configuration élémentaire du pare-feu* du *Guide de personnalisation de Red Hat Linux*.

4.1.29. /etc/sysconfig/redhat-config-users

Le fichier /etc/sysconfig/redhat-config-users est le fichier de configuration pour l'application graphique **Gestionnaire d'utilisateurs**. Sous Red Hat Linux 9 ce fichier est utilisé pour filtrer les utilisateurs du système tels que root, démon ou lp. Ce fichier peut être édité depuis le menu déroulant **Préférences => Filtrer les utilisateurs du système et groupes** dans l'application **Gestionnaire d'utilisateurs** et ne doit pas être modifié manuellement. Pour de plus amples informations sur l'utilisation de cette application, consultez le chapitre intitulé *Configuration des utilisateurs et des groupes* du *Guide de personnalisation de Red Hat Linux*.

4.1.30. /etc/sysconfig/redhat-logviewer

Le fichier /etc/sysconfig/redhat-logviewer est le fichier de configuration pour l'application d'affichage de journal graphique et interactive, **Afficheur de journal**. Ce fichier peut être édité depuis le menu déroulant **Éditer => Préférences** dans l'application **Afficheur de journal** application et ne doit pas être modifié manuellement. Pour de plus amples informations sur l'utilisation de cette application, consultez le chapitre intitulé *Fichiers journaux* du *Guide de personnalisation de Red Hat Linux*.

4.1.31. /etc/sysconfig/samba

Le fichier /etc/sysconfig/samba est utilisé pour transmettre des arguments aux démons smbd et nmbd au moment du démarrage. Le démon smbd offre une connectivité de partage de fichiers pour les clients Windows sur le réseau. Le démon nmbd offre NetBIOS sur les services de nommage IP. Pour de plus amples informations sur les paramètres pouvant être utilisés dans ce fichier, consultez la page

de manuel relative à smbd. Par défaut, ce fichier règle le fonctionnement de smbd et nmbd en mode démon.

4.1.32. /etc/sysconfig/sendmail

Le fichier /etc/sysconfig/sendmail permet d'envoyer des messages à un ou plusieurs destinataires, en acheminant les messages sur les réseaux nécessaires, quels qu'ils soient. Le fichier définit les valeurs par défaut pour l'exécution de l'application /etc/sysconfig/sendmail. En raison de ses valeurs par défaut, il s'exécute comme démon en tâche de fond et qu'il contrôle sa file d'attente une fois par heure, si quelque chose a été sauvegardé.

Les valeurs suivantes peuvent être utilisées:

- DAEMON=<*valeur*>, où <*valeur*> correspond à une des valeurs booléennes suivantes:
  - yes — **Sendmail** doit être configuré pour contrôler le port 25 afin de détecter le courrier entrant. La valeur yes implique l'utilisation des options -bd.
  - no — **Sendmail** ne doit pas être configuré pour contrôler le port 25 afin de détecter le courrier entrant.
- QUEUE=1h qui est donné à **Sendmail** en tant que -q$QUEUE. L'option -q n'est pas donnée à **Sendmail** si le fichier /etc/sysconfig/sendmail existe et que QUEUE est vide ou non-défini.

4.1.33. /etc/sysconfig/soundcard

Le fichier /etc/sysconfig/soundcard est créé par sndconfig et ne devrait pas être modifié. Le seul rôle de ce fichier est de déterminer l'entrée de carte du menu à afficher par défaut lors de la prochaine exécution de sndconfig. Les informations de configuration de la carte son se trouvent dans le fichier /etc/modules.conf.

Ce dernier peut contenir les éléments suivants:

- CARDTYPE=<*valeur*>, où <*valeur*> est réglée par exemple, sur SB16 pour une carte son Soundblaster 16.

4.1.34. /etc/sysconfig/spamassassin

Le fichier /etc/sysconfig/spamassassin est utilisé pour transmettre des arguments au démon spamd (une version 'démonisée' de Spamassassin) lors du démarrage. Spamassassin est une application de messagerie pour le filtrage de pourriel. ('spam'). Pour obtenir une liste des options disponibles, consultez la page de manuel relative à spamd. Par défaut, il configure spamd de sorte à ce qu'il s'exécute en mode démon, crée des préférences utilisateur et crée automatiquement des listes blanches.

Pour de plus amples informations sur Spamassassin, consultez la Section 11.4.2.6.

4.1.35. /etc/sysconfig/squid

Le fichier /etc/sysconfig/squid est utilisé pour transmettre des arguments au démon squid au moment du démarrage. Le démon squid est un serveur proxy de cache pour des applications clientes par le Web. Pour de plus amples informations sur la configuration d'un serveur proxy squid, ouvrez le répertoire /usr/share/doc/squid-<*version*>/ à l'aide de votre navigateur (remplacez

<*version*> par le numéro de la version `squid` installée sur votre système). Par défaut, ce fichier règle le démarrage premier de `squid` en mode démon et le délai avant une interruption automatique.

4.1.36. /etc/sysconfig/tux

Le fichier `/etc/sysconfig/tux` est le fichier de configuration de 'Red Hat Content Accelerator' (précédemment appelé TUX), le serveur Web basé sur le noyau. Pour de plus amples informations sur la configuration de Red Hat Content Accelerator, ouvrez le répertoire `/usr/share/doc/tux-`<*version*>`/tux/index.html` à l'aide de votre navigateur (remplacez <*version*> par le numéro de la version de TUX installée sur votre système). Les paramètres disponibles pour ce fichier sont énumérés dans `/usr/share/doc/tux-`<*version*>`/tux/parameters.html`.

4.1.37. /etc/sysconfig/ups

Le fichier `/etc/sysconfig/ups` est utilisé pour spécifier les informations relatives à tout *système d'alimentation ininterrompue* (ou *UPS* de l'anglais 'Uninterruptible Power Supplies') branché au système. Un UPS peut être très utile à un système Red Hat Linux car il donne le temps nécessaire pour éteindre l'ordinateur correctement lors d'une panne de courant. Les valeurs suivantes peuvent être utilisées:

- `SERVER=`<*valeur*>, où <*valeur*> correspond à l'un des éléments suivants:
  - `yes` — un dispositif UPS est branché à votre système.
  - `no` — aucun dispositif UPS n'est branché à votre système.

- `MODEL=`<*valeur*>, où <*valeur*> doit correspondre à un des éléments suivants ou doit être réglée sur `NONE` (aucun) si aucun dispositif UPS n'est branché au système:
  - `apcsmart` — un périphérique APC SmartUPS™ ou semblable.
  - `fentonups` — un dispositif Fenton UPS™.
  - `optiups` — un dispositif OPTI-UPS™.
  - `bestups` — un dispositif Best Power™.
  - `genericups` — un dispositif UPS générique.
  - `ups-trust425+625` — un dispositif UPS Trust™.

- `DEVICE=`<*valeur*>, où <*valeur*> spécifie où le dispositif UPS est branché, comme par exemple `/dev/ttyS0`.
- `OPTIONS=`<*valeur*>, où <*valeur*> correspond à une commande spéciale qui doit être passée au dispositif UPS.

4.1.38. /etc/sysconfig/vncservers

Le fichier `/etc/sysconfig/vncservers` configure la façon dont le serveur '*Virtual Network Computing*' (ou *VNC*) démarre.

VNC est un système d'affichage à distance qui vous permet de visualiser un environnement bureau non seulement sur l'ordinateur où il est exécuté mais également sur différents réseau d'architectures variées.

Ce fichier peut contenir les éléments suivants:

- `VNCSERVERS=<valeur>`, où `<valeur>` est réglée sur une valeur ressemblant à `"1:fred"`, pour indiquer qu'un serveur VNC devrait être démarré pur l'utilisateur fred sur l'écran :1. L'utilisateur fred doit avoir configuré un mot de passe VNC en utilisant `vncpasswd` avant d'essayer de se connecter au serveur VNC distant.

Remarquez bien que lors de l'utilisation d'un serveur VNC, la communication que vous établissez avec le serveur n'est pas cryptée. Pour cette raison, il est vivement déconseillé de l'utiliser sur un réseau à faible sécurité. Pour des instructions spécifiques sur l'utilisation de SSH pour sécuriser la communication avec le serveur VNC, lisez les informations présentes sur le site http://www.uk.research.att.com/vnc/sshvnc.html. Pour de plus amples informations sur SSH, reportez-vous au Chapitre 18 ou au *Guide de personnalisation de Red Hat Linux*.

4.1.39. `/etc/sysconfig/xinetd`

Le fichier `/etc/sysconfig/xinetd` est utilisé pour transmettre des arguments au démon `xinetd` au moment du démarrage. Le démon `xinetd` lance des programmes qui fournissent des services Internet lorsqu'une requête est reçue sur le port de ce service. Pour des plus amples informations sur les paramètres que vous pouvez utiliser dans ce fichier, consultez la page de manuel relative à `xinetd`. Pour des plus amples informations sur le service `xinetd`, consultez la Section 15.3.

4.2. Répertoires contenus dans le répertoire `/etc/sysconfig/`

Les répertoires suivants se trouvent normalement dans `/etc/sysconfig/`.

- `apm-scripts` — contient le script Red Hat APM suspendre/reprendre. Il est déconseillé d'éditer directement ce fichier. Si vous devez le personnaliser, il suffit de créer un fichier nommé `/etc/sysconfig/apm-scripts/apmcontinue` et il sera invoqué à la fin du script. Vous pouvez aussi contrôler le script en éditant `/etc/sysconfig/apmd`.
- `cbq` — ce répertoire contient les fichiers de configuration nécessaires pour le 'Class Based Queuing' (rangement selon la classe) pour la gestion de la largeur de bande sur les interfaces réseau.
- `networking` — ce répertoire est utilisé par l'**Outil d'administration de réseau** (`redhat-config-network`) et son contenu ne devrait pas être modifié manuellement. Pour de plus amples informations sur la configuration des interfaces réseau à l'aide de l'utilitaire **Outil d'administration de réseau**, consultez le chapitre intitulé *Configuration réseau* du *Guide de personnalisation de Red Hat Linux*.
- `network-scripts` — ce répertoire contient les fichiers de configuration relatifs au réseau ci-dessous:
  - Les fichiers de configuration réseau pour chaque interface réseau configurée, comme par exemple, `ifcfg-eth0` pour l'interface Ethernet `eth0`.
  - Les scripts utilisés pour activer et désactiver des interfaces réseau, comme par exemple, `ifup` et `ifdown`.
  - Les scripts utilisés pour activer et désactiver des interfaces réseau ISDN, comme par exemple, `ifup-isdn` et `ifdown-isdn`
  - Divers scripts de fonctions réseau partagés, qu'il est vivement déconseillé de modifier directement.

  Pour de plus amples informations sur le répertoire `network-scripts`, consultez le Chapitre 8.
- `rhn` — ce répertoire contient les fichiers de configuration ainsi que les clés GPG pour Red Hat Network. Aucun fichier de ce répertoire ne devrait être éditer manuellement. Pour de plus amples informations sur Red Hat Network, consultez son site Web à l'adresse suivante: https://rhn.redhat.com.

4.3. Ressources supplémentaires

L'intention de ce chapitre est seulement de fournir une introduction aux fichiers contenus dans le répertoire `/etc/sysconfig/`. Les sources ci-dessous contiennent des informations plus détaillées.

4.3.1. Documentation installée

- `/usr/share/doc/initscripts-`<*numéro-version*>`/sysconfig.txt` — Ce fichier contient une liste plus complète des fichiers se trouvant dans le répertoire `/etc/sysconfig/` et des options qu'ils acceptent. Le <*numéro-version*> dans le chemin d'accès vers ce fichier correspond à la version du paquetage `initscripts` installée.



Chapitre 5.
Le système de fichiers proc

Le noyau de Linux a deux fonctions principales: contrôler l'accès aux périphériques physiques de l'ordinateur d'une part et programmer à quel moment et de quelle façon les processus vont interagir avec ces périphériques d'autre part. Le répertoire `/proc/` contient une hiérarchie de fichiers spéciaux qui représentent l'état actuel du noyau; cela permet aux applications ainsi qu'aux utilisateurs de scruter la perception du noyau du système.

Vous pouvez trouver dans le répertoire `/proc/` de nombreuses informations relatives à la configuration matérielle du système et aux processus en cours d'exécution. De plus, certains des fichiers situés dans l'arborescence du répertoire `/proc/` peuvent être manipulés par les utilisateurs ainsi que par les applications afin de transmettre des changements de configuration au noyau.

5.1. Un système de fichiers virtuel

Sous Linux, toutes les données sont stockées en tant que fichiers. La plupart des utilisateurs sont familiers avec les deux principaux types de fichiers: texte et binaire. Mais le répertoire `/proc/` contient un autre type de fichier nommé *fichier virtuel*. C'est pour cette raison que `/proc/` est souvent désigné sous le nom de *système de fichiers virtuel*.

Ces fichiers virtuels ont des qualités uniques. La plupart d'entre eux ont une taille égale à zéro octet; pourtant, lorsqu'on les affiche, on constate qu'ils contiennent parfois une grande quantité d'informations. De plus, la plupart du temps, les paramètres date et heure des fichiers virtuels reflètent la date et l'heure actuelles et montrent ainsi le fait qu'ils sont mis à jour continuellement.

Des fichiers virtuels tels que `/proc/interrupts`, `/proc/meminfo`, `/proc/mounts`,et `/proc/partitions` fournissent un aperçu de l'environnement d'un système à un moment donné. D'autres tels que `/proc/filesystems` et le répertoire `/proc/sys/` fournissent des informations sur la configuration du système ainsi que des interfaces.

À des fins d'organisation, les fichiers qui contiennent des informations sur un sujet similaire sont groupés dans des répertoires et sous-répertoires virtuels. Par exemple, `/proc/ide/` contient des informations se rapportant à tous les périphériques IDE. De même, les répertoires 'process' contiennent des données concernant tous les processus en cours d'exécution sur le système.

5.1.1. Affichage de fichiers virtuels

En appliquant les commandes `cat`, `more` ou `less` aux fichiers du répertoire `/proc/`, vous avez immédiatement accès à une très importante source d'informations sur le système. Par exemple, pour connaître le type d'unité centrale dont dispose votre ordinateur, tapez `cat /proc/cpuinfo` et vous obtiendrez une sortie semblable à l'extrait ci-dessous:

```
processor : 0
vendor_id : AuthenticAMD
cpu family : 5
model  : 9
model name : AMD-K6(tm) 3D+ Processor
stepping : 1
cpu MHz  : 400.919
cache size : 256 KB
fdiv_bug : no
hlt_bug  : no
f00f_bug : no
coma_bug : no
```

```
fpu  : yes
fpu_exception : yes
cpuid level : 1
wp  : yes
flags  : fpu vme de pse tsc msr mce cx8 pge mmx syscall 3dnow k6_mtrr
bogomips : 799.53
```

Lorsque vous affichez différents fichiers virtuels dans le système de fichiers `/proc/` vous pouvez remarquer que certaines des informations sont facilement compréhensibles tandis que d'autres sont codées. C'est en partie pour cela qu'existent des utilitaires dont la fonction consiste à récupérer des données de fichiers virtuels et de les afficher de façon compréhensible. Parmi ces utilitaires figurent par exemple: `lspci`, `apm`, `free`, et `top`.



Remarque

Certains des fichiers virtuels du répertoire `/proc/` ne peuvent être lus que par l'utilisateur root.

5.1.2. Modification de fichiers virtuels

En général, la plupart des fichiers virtuels du répertoire `/proc/` sont en lecture-seule. Certains peuvent toutefois être utilisés pour régler les paramètres dans le noyau. Cela vaut particulièrement pour les fichiers du sous-répertoire `/proc/sys/`.

Pour modifier la valeur d'un fichier virtuel, utilisez la commande `echo` et un symbole > afin de réacheminer la nouvelle valeur vers le fichier. Par exemple, pour modifier votre nom d'hôte rapidement, vous pouvez taper:

```
echo www.example.com > /proc/sys/kernel/hostname
```

D'autres fichiers servent de commutateur binaire ou booléen. Par exemple, si vous tapez `cat /proc/sys/net/ipv4/ip_forward` vous obtiendrez comme sortie un `0` ou un `1`. Le `0` indique que le noyau ne réachemine pas les paquets réseau. En utilisant la commande `echo` pour modifier la valeur du fichier `ip_forward` en `1`, vous pouvez déclencher immédiatement le réacheminement des paquets.



Astuce

La commande `/proc/sys/` permet également de modifier les paramètres du sous-répertoire `/sbin/sysctl`. Pour obtenir davantage d'informations sur cette commande, reportez-vous à la Section 5.4

Pour obtenir une liste de certains des fichiers de configuration du noyau disponibles dans `/proc/sys/`, consultez la Section 5.3.9.

5.2. Les fichiers du niveau supérieur dans le système de fichiers proc

Ci-dessous figure une liste de certains des fichiers virtuels les plus utiles du niveau supérieur du répertoire `/proc/`.



Remarque

Dans la plupart des cas, le contenu des fichiers répertoriés dans cette section sera différent sur votre ordinateur. En effet, une bonne partie des informations dépendent de la configuration matérielle sur laquelle vous exécutez Red Hat Linux.

5.2.1. /proc/apm

Ce fichier fournit des informations sur l'état du système de *gestion de la consommation d'énergie (APM)* (de l'anglais 'Advanced Power Management') et est utilisé par la commande `apm`. Si le système sans batterie est connecté à une source d'alimentation de courant alternatif, ce fichier virtuel sera similaire à:

```
1.16 1.2 0x07 0x01 0xff 0x80 -1% -1 ?
```

La sortie résultant de l'exécution de la commande `apm -v` ressemble à l'extrait ci-dessous:

```
APM BIOS 1.2 (kernel driver 1.16)
AC on-line, no system battery
```

Pour les systèmes n'utilisant pas de batterie comme source d'alimentation, `apm` ne peut guère faire grand chose de plus que de mettre l'ordinateur en mode veille. La commande `apm` est beaucoup plus utile sur les portables. Ci-dessous se trouve l'exemple d'une sortie résultant de l'exécution de la commande `cat /proc/apm` sur un portable utilisant Red Hat Linux lorsqu'il est branché à une prise de courant:

```
1.16 1.2 0x03 0x01 0x03 0x09 100% -1 ?
```

Si l'on débranche ce portable de sa source d'alimentation pendant quelques minutes, le contenu du fichier `apm` change de la manière suivante:

```
1.16 1.2 0x03 0x00 0x00 0x01 99% 1792 min
```

La commande `apm -v` va à présent générer des données plus utiles, comme par exemple:

```
APM BIOS 1.2 (kernel driver 1.16)
AC off-line, battery status high: 99% (1 day, 5:52)
```

5.2.2. /proc/cmdline

Ce fichier montre les paramètres transmis au noyau au moment du démarrage. Un exemple de fichier `/proc/cmdline` ressemble à l'exemple ci-dessous:

```
ro root=/dev/hda2
```

Cet extrait nous indique que le noyau est monté en lecture-seule (comme l'indique l'élément `(ro)` signifiant 'read only') sur la deuxième partition du premier périphérique IDE (`/dev/hda2`).

5.2.3. /proc/cpuinfo

Ce fichier virtuel identifie le type de processeur utilisé par votre système. L'extrait ci-dessous montre un exemple de la sortie typique de `/proc/cpuinfo`:

```
processor       : 0
vendor_id       : AuthenticAMD
cpu family      : 5
model           : 9
model name      : AMD-K6(tm) 3D+ Processor
stepping        : 1
cpu MHz         : 400.919
cache size      : 256 KB
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level     : 1
wp              : yes
flags           : fpu vme de pse tsc msr mce cx8 pge mmx syscall 3dnow k6_mtrr
bogomips        : 799.53
```

- `processor` — Fournit à chaque processeur un numéro d'identification. Sur les systèmes dotés d'un seul processeur, ce numéro sera `0`.
- `cpu family` — Identifie avec certitude le type de processeur dont votre système dispose. Si vous disposez d'un système Intel, placez simplement ce numéro devant "86" afin de déterminer la valeur. Cela est particulièrement utile si vous essayez d'identifier l'architecture d'un système plus ancien (586, 486 ou 386). Comme certains paquetages RPM sont compilés pour chacune de ces architectures particulières, cette valeur vous indique également quel paquetage installer.
- `model name` — Affiche le nom communément utilisé du processeur, de même que son nom de projet.
- `cpu MHz` — Indique la vitesse précise en mégahertz du processeur (au millième près).
- `cache size` — Indique la quantité de mémoire cache de niveau 2 disponible pour le processeur.
- `flags` — Définit un certain nombre de caractéristiques du processeur, telle que la présence d'une unité de virgule flottante ou FPU ('Floating Point Unit') et la capacité à traiter des instructions MMX.

5.2.4. /proc/devices

Ce fichier affiche les divers périphériques d'entrée-sortie de caractères et blocs actuellement configurés (il ne contient pas les périphériques dont les modules ne sont pas chargés). Ci-dessous figure un exemple de ce fichier virtuel:

```
Character devices:
  1 mem
  2 pty
  3 ttyp
  4 ttyS
  5 cua
  7 vcs
 10 misc
 14 sound
 29 fb
 36 netlink
128 ptm
```

```
129 ptm
136 pts
137 pts
162 raw
254 iscsictl

Block devices:
  1 ramdisk
  2 fd
  3 ide0
  9 md
 22 ide1
```

La sortie de `/proc/devices` inclut le nombre ainsi que le nom principal du périphérique; elle est répartie en deux sections principales: `Character devices` (périphériques d'entrée-sortie de caractères) et `Block devices` (périphériques blocs).

Les *périphériques d'entrée-sortie de caractères* sont semblables aux *périphériques blocs*, à l'exception de deux points essentiels:

1. Les périphériques blocs ont un tampon disponible, ce qui leur permet de classer les demandes avant de les traiter. Cela est très important dans le cas des périphériques conçus pour stocker des informations — tels que les disques durs — parce que la possibilité de classer les informations avant de les écrire sur le périphérique permet de les placer de façon plus efficace. Les périphériques d'entrée-sortie de caractères ne nécessitent pas de tamponnement.
2. L'autre différence réside dans le fait que les périphériques blocs peuvent envoyer et recevoir les informations par blocs de taille spécifique, configurée par périphérique. Les périphériques d'entrée-sortie de caractères envoient des données sans taille préconfigurée.

Vous trouverez plus d'informations sur les périphériques dans `/usr/src/linux-2.4/Documentation/devices.txt`.

5.2.5. /proc/dma

Ce fichier contient une liste des canaux ISA d'accès direct à la mémoire (ADM) enregistrés qui sont utilisés. Un exemple de fichier `/proc/dma` ressemble l'exemple ci-dessous:

```
 4: cascade
```

5.2.6. /proc/execdomains

Ce fichier donne la liste des *domaines d'exécution* actuellement pris en charge par le noyau Linux, ainsi que la gamme des personnalités qu'ils prennent en charge.

```
0-0   Linux           [kernel]
```

Considérez les domaines d'exécution comme étant la "personnalité" d'un système d'exploitation donné. Parce que d'autres formats binaires, tels que Solaris, UnixWare et FreeBSD peuvent être utilisés avec Linux, les programmeurs peuvent, en changeant la personnalité d'une tâche, changer la façon dont le système d'exploitation traite certains appels système de ces binaires. À l'exception du domaine d'exécution `PER_LINUX`, différentes personnalités peuvent être mises en oeuvre en tant que modules chargeables de façon dynamique.

5.2.7. /proc/fb

Ce fichier contient une liste des périphériques de mémoires vidéo, comportant le numéro de chaque périphérique et le pilote qui le contrôle. La sortie de `/proc/fb` pour les systèmes qui contiennent des périphériques de mémoire vidéo ressemble généralement l'exemple ci-dessous:

```
0 VESA VGA
```

5.2.8. /proc/filesystems

Ce fichier affiche une liste des types de systèmes de fichiers actuellement pris en charge par le noyau. Ci-dessous figure un exemple de sortie d'un fichier `/proc/filesystems` générique:

```
nodev rootfs
nodev bdev
nodev proc
nodev sockfs
nodev tmpfs
nodev shm
nodev pipefs
 ext2
nodev ramfs
 iso9660
nodev devpts
 ext3
nodev autofs
nodev binfmt_misc
```

La première colonne indique si le système de fichiers est monté sur un périphérique bloc. Ceux commençant par `nodev` ne sont pas montés sur un périphérique. La seconde colonne répertorie les noms de systèmes de fichiers pris en charge.

La commande `mount` tourne en boucle dans ces systèmes de fichiers lorsque aucun d'eux n'est spécifié comme argument.

5.2.9. /proc/interrupts

Ce fichier enregistre le nombre d'interruptions par IRQ sur l'architecture x86. Un fichier `/proc/interrupts` standard ressemble à l'extrait suivant:

```
           CPU0
  0:   80448940          XT-PIC  timer
  1:     174412          XT-PIC  keyboard
  2:          0          XT-PIC  cascade
  8:          1          XT-PIC  rtc
 10:     410964          XT-PIC  eth0
 12:      60330          XT-PIC  PS/2 Mouse
 14:    1314121          XT-PIC  ide0
 15:    5195422          XT-PIC  ide1
NMI:          0
ERR:          0
```

Dans le cas d'un ordinateur ayant plusieurs processeurs, le fichier peut être légèrement différent:

```
           CPU0       CPU1
  0: 1366814704          0          XT-PIC  timer
```

```
  1:        128        340    IO-APIC-edge  keyboard
  2:          0          0          XT-PIC  cascade
  8:          0          1    IO-APIC-edge  rtc
 12:       5323       5793    IO-APIC-edge  PS/2 Mouse
 13:          1          0          XT-PIC  fpu
 16:   11184294   15940594   IO-APIC-level  Intel EtherExpress Pro 10/100
Ethernet
 20:    8450043   11120093   IO-APIC-level  megaraid
 30:      10432      10722   IO-APIC-level  aic7xxx
 31:         23         22   IO-APIC-level  aic7xxx
NMI:          0
ERR:          0
```

La première colonne fait référence au numéro IRQ. Chaque unité centrale du système a sa propre colonne et son propre nombre d'interruptions par IRQ. La colonne suivante indique le type d'interruption et la dernière colonne contient le nom du périphérique situé à cet IRQ.

Chaque type d'interruption, spécifique à l'architecture, qui présenté dans ce fichier, a une signification légèrement différente. Pour les ordinateurs x86, les valeurs suivantes sont courantes:

- `XT-PIC` — Il s'agit des anciennes interruptions des ordinateurs AT.
- `IO-APIC-edge` — Signal de voltage sur ces transitions d'interruption de faible à élevé, créant une *dénivellation*, là où l'interruption a lieu; il n'est signalé qu'une seule fois. Des interruptions de ce genre, de même que l'interruption `IO-APIC-level`, ne se rencontrent que sur des systèmes ayant des processeurs de la gamme 586 ou d'une gamme supérieure.
- `IO-APIC-level` — Génère des interruptions lorsque le signal de voltage devient élevé, jusqu'à ce qu'il redevienne faible.

5.2.10. /proc/iomem

Ce fichier montre la topographie mémoire actuelle du système pour chacun de ses périphériques physiques:

```
00000000-0009fbff : System RAM
0009fc00-0009ffff : reserved
000a0000-000bffff : Video RAM area
000c0000-000c7fff : Video ROM
000f0000-000fffff : System ROM
00100000-07ffffff : System RAM
  00100000-00291ba8 : Kernel code
  00291ba9-002e09cb : Kernel data
e0000000-e3ffffff : VIA Technologies, Inc. VT82C597 [Apollo VP3]
e4000000-e7ffffff : PCI Bus #01
  e4000000-e4003fff : Matrox Graphics, Inc. MGA G200 AGP
  e5000000-e57fffff : Matrox Graphics, Inc. MGA G200 AGP
e8000000-e8ffffff : PCI Bus #01
  e8000000-e8ffffff : Matrox Graphics, Inc. MGA G200 AGP
ea000000-ea00007f : Digital Equipment Corporation DECchip 21140 [FasterNet]
  ea000000-ea00007f : tulip
ffff0000-ffffffff : reserved
```

La première colonne affiche les registres de mémoire utilisés par chacun des différents types de mémoire. La seconde colonne indique le type de mémoire situé dans ces registres. Cette colonne vous indique notamment quels registres sont utilisés par le noyau dans la mémoire vive du système ou, si vous avez plusieurs ports Ethernet sur votre carte d'interface réseau, les registres de mémoire affectés à chaque port.

5.2.11. /proc/ioports

La sortie de /proc/ioports fournit une liste des fourchettes relatives aux ports actuellement enregistrés et utilisés pour les communications d'entrée et de sortie avec un périphérique. Ce fichier, qui peut être assez long, affiche un début semblable à ce qui suit:

```
0000-001f : dma1
0020-003f : pic1
0040-005f : timer
0060-006f : keyboard
0070-007f : rtc
0080-008f : dma page reg
00a0-00bf : pic2
00c0-00df : dma2
00f0-00ff : fpu
0170-0177 : ide1
01f0-01f7 : ide0
02f8-02ff : serial(auto)
0376-0376 : ide1
03c0-03df : vga+
03f6-03f6 : ide0
03f8-03ff : serial(auto)
0cf8-0cff : PCI conf1
d000-dfff : PCI Bus #01
e000-e00f : VIA Technologies, Inc. Bus Master IDE
  e000-e007 : ide0
  e008-e00f : ide1
e800-e87f : Digital Equipment Corporation DECchip 21140 [FasterNet]
  e800-e87f : tulip
```

La première colonne indique la plage d'adresses de port E/S réservées au périphérique spécifié dans la seconde colonne.

5.2.12. /proc/isapnp

Ce fichier répertorie les cartes *Plug & Play (PnP)* installées dans les connecteurs ISA du système. Cela se voit surtout avec les cartes son, mais peut aussi inclure tout autre périphérique. Un fichier /proc/isapnp ayant une entrée Soundblaster ressemble à l'extrait suivant:

```
Card 1 'CTL0070:Creative ViBRA16C PnP' PnP version 1.0 Product version 1.0
  Logical device 0 'CTL0001:Audio'
    Device is not active
    Active port 0x220,0x330,0x388
    Active IRQ 5 [0x2]
    Active DMA 1,5
    Resources 0
      Priority preferred
      Port 0x220-0x220, align 0x0, size 0x10, 16-bit address decoding
      Port 0x330-0x330, align 0x0, size 0x2, 16-bit address decoding
      Port 0x388-0x3f8, align 0x0, size 0x4, 16-bit address decoding
      IRQ 5 High-Edge
      DMA 1 8-bit byte-count compatible
      DMA 5 16-bit word-count compatible
      Alternate resources 0:1
        Priority acceptable
        Port 0x220-0x280, align 0x1f, size 0x10, 16-bit address decoding
        Port 0x300-0x330, align 0x2f, size 0x2, 16-bit address decoding
```

```
Port 0x388-0x3f8, align 0x0, size 0x4, 16-bit address decoding
IRQ 5,7,2/9,10 High-Edge
DMA 1,3 8-bit byte-count compatible
DMA 5,7 16-bit word-count compatible
```

Ce fichier peut être assez long, tout dépend du nombre de périphériques affichés et de leurs besoins en ressources.

Chaque carte affiche son nom, le numéro de version PnP et le numéro de version du produit. Si le périphérique est activé et configuré, ce fichier indique également le port ainsi que les numéros IRQ pour le périphérique. De plus, afin d'assurer une meilleure compatibilité, la carte spécifie les valeurs `preferred` (préférable) et `acceptable` pour un certain nombre de paramètres. L'objectif est de permettre aux cartes PnP de fonctionner conjointement et d'éviter tout conflit d'IRQ ou de port.

5.2.13. /proc/kcore

Ce fichier représente la mémoire physique du système et est stocké sous forme de fichier 'core'. Contrairement à la plupart des fichiers `/proc/`, le fichier `kcore` affiche une taille. Cette valeur est donnée en octets et est égale à la taille de la mémoire vive (RAM) utilisée plus 4 Ko.

Le contenu de ce fichier, conçu pour être examiné par un débogueur, tel que `gdb`, est codé.

**Avertissement**

N'affichez pas le fichier virtuel `/proc/kcore`. Le contenu de ce fichier va submerger votre terminal de texte. Si vous l'ouvrez par accident, appuyez sur les touches [Ctrl]-[C] pour arrêter le processus, puis tapez `reset` pour faire revenir l'invite de ligne de commande.

5.2.14. /proc/kmsg

Ce fichier est utilisé pour contenir des messages générés par le noyau. Ces messages sont ensuite récupérés par d'autres programmes, tels que `/sbin/klogd`.

5.2.15. /proc/ksyms

Ce fichier contient les définitions des symboles utilisées par les outils de modules pour lier et associer dynamiquement des modules du noyau.

```
e003def4 speedo_debug [eepro100]
e003b04c eepro100_init [eepro100]
e00390c0 st_template [st]
e002104c RDINDOOR [megaraid]
e00210a4 callDone [megaraid]
e00226cc megaraid_detect [megaraid]
```

La première colonne indique l'adresse de la mémoire pour la fonction du noyau, la deuxième colonne fait référence au nom de la fonction et la dernière donne le nom du module chargé.

5.2.16. /proc/loadavg

Ce fichier fournit un aperçu de la moyenne de charge sur le processeur ainsi que des données supplémentaires utilisées par la commande `uptime` ainsi que par d'autres commandes. Ci-dessous figure un exemple de ce à quoi un fichier `/proc/loadavg` peut ressembler:

```
0.20 0.18 0.12 1/80 11206
```

Les trois premières colonnes mesurent l'utilisation de l'unité centrale en fonction des dernières périodes de 1, 5 et 10 minutes. La quatrième colonne indique le nombre de processus en cours d'exécution ainsi que le nombre total de processus. La dernière colonne affiche le dernier ID de processus utilisé.

5.2.17. /proc/locks

Ce fichier affiche les fichiers actuellement verrouillés par le noyau. Le contenu de ce fichier comprend des données internes de débogage du noyau et peut varier énormément en fonction de l'utilisation du système. Ci-après figure un exemple de fichier `/proc/locks` d'un système peu chargé:

```
1: FLOCK  ADVISORY  WRITE 807 03:05:308731 0 EOF c2a260c0 c025aa48 c2a26120
2: POSIX  ADVISORY  WRITE 708 03:05:308720 0 EOF c2a2611c c2a260c4 c025aa48
```

Chaque verrouillage a sa propre ligne qui commence par un numéro unique. La deuxième colonne indique la classe de verrouillage utilisée dans laquelle `FLOCK` représente les verrouillages de fichiers UNIX de type plus ancien d'un appel système `flock` et `POSIX` représente les verrouillages POSIX plus récents, de l'appel système `lockf`.

La troisième colonne peut avoir 2 valeurs: `ADVISORY` ou `MANDATORY`. La valeur `ADVISORY` signifie que le verrouillage n'empêche pas les autres personnes d'avoir accès aux données; il ne fait que les empêcher d'essayer de les verrouiller. La valeur `MANDATORY` quant à elle, signifie que personne d'autre n'est autorisé à accéder aux données tant que le verrouillage est en effectif. La quatrième colonne indique si le verrouillage autorise le détenteur à avoir un accès `READ` (lecture) ou `WRITE` (écriture) au fichier et la cinquième colonne montre l'identifiant du processus qui détient le verrouillage. La sixième colonne montre l'identifiant du fichier verrouillé, selon le format suivant: *PÉRIPHÉRIQUE-PRINCIPAL*:*PÉRIPHÉRIQUE-MINEUR*:*NUMÉRO-INODE*. La septième colonne indique le début et la fin de la région verrouillée du fichier. Les autres colonnes pointent vers des structures de données internes du noyau utilisées à des fins de débogage spécialisé et peuvent être ignorées.

5.2.18. /proc/mdstat

Ce fichier contient des informations sur les configurations RAID à disques multiples. Si votre système ne dispose pas de ce genre de configuration, votre fichier `/proc/mdstat` ressemblera à l'extrait suivant:

```
Personalities :
read_ahead not set
unused devices: <none>
```

Ce fichier garde l'état reproduit ci-dessus, sauf si vous créez un périphérique logiciel RAID ou `md`. Dans ce cas, vous pouvez afficher `/proc/mdstat` pour avoir une idée générale du status actuel de vos périphériques RAID `md`*X*.

Le fichier `/proc/mdstat` ci-dessous montre un système contenant `md0`configuré comme un périphérique RAID 1 et effectuant la re-synchronisation des disques:

```
Personalities : [linear] [raid1]
```

```
read_ahead 1024 sectors
md0: active raid1 sda2[1] sdb2[0] 9940 blocks [2/2] [UU] resync=1%
finish=12.3min
algorithm 2 [3/3] [UUU]
unused devices: <none>
```

5.2.19. /proc/meminfo

Ci-dessous figure l'un des fichiers les plus communément utilisés du répertoire `/proc/` car il donne de nombreuses informations importantes sur l'utilisation de la mémoire vive du système.

Un système ayant 256 Mo de mémoire vive et 384 Mo d'espace swap pourrait avoir un fichier `/proc/meminfo` semblable à celui-ci:

```
        total:    used:    free:  shared: buffers:  cached:
Mem:  261709824 253407232  8302592        0 120745984 48689152
Swap: 402997248     8192 402989056
MemTotal:       255576 kB
MemFree:          8108 kB
MemShared:           0 kB
Buffers:        117916 kB
Cached:          47548 kB
Active:         135300 kB
Inact_dirty:     29276 kB
Inact_clean:       888 kB
Inact_target:        0 kB
HighTotal:           0 kB
HighFree:            0 kB
LowTotal:       255576 kB
LowFree:          8108 kB
SwapTotal:      393552 kB
SwapFree:       393544 kB
```

La plupart des informations de cet exemple sont utilisées par les commandes `free`, `top` et `ps`. En fait, la sortie de la commande `free` est même similaire en apparence au contenu et à la structure de `/proc/meminfo`. Mais si vous examinez directement `/proc/meminfo`, vous y trouverez davantage d'informations:

- `Mem` — Affiche l'état courant de la mémoire vive du système, y compris une répartition détaillés de l'utilisation en octets des mémoires totale, utilisée, libre, partagée, tampon et cache.
- `Swap` — Affiche la quantité totale, utilisée et libre d'espace swap, en octets.
- `MemTotal` — Quantité totale de mémoire vive (exprimée en Ko).
- `MemFree` — Quantité de mémoire vive (exprimée en Ko), non utilisée par le système.
- `MemShared` — Non utilisé avec les noyaux 2.4 ou supérieurs, mais gardé pour des raisons de compatibilité avec les versions de noyau précédentes.
- `Buffers` — Quantité de mémoire vive (exprimée en Ko), utilisée pour les tampons de fichiers.
- `Cached` — Quantité de mémoire vive (exprimée en Ko), utilisée comme mémoire cache.
- `Active` — Quantité totale de mémoire tampon ou de mémoire cache de pages (exprimée en Ko), en utilisation active.
- `Inact_dirty` — Quantité totale de tampon ou de pages de cache (exprimée en Ko), qui peut être libre et disponible.
- `Inact_clean` — Quantité totale de tampon ou de pages de cache (exprimée en Ko), qui est réellement libre et disponible.

- `Inact_target` — La quantité nette d'allocations par seconde (exprimée en Ko), en moyenne par minute.
- `HighTotal` et `HighFree` — Respectivement la quantité totale et libre de mémoire qui n'est pas directement mappée dans l'espace du noyau. La valeur `HighTotal` peut varier en fonction du type de noyau utilisé.
- `LowTotal` et `LowFree` — Respectivement, la quantité totale et libre de mémoire qui est directement mappée dans l'espace du noyau. La valeur `LowTotal` peut varier en fonction du type de noyau utilisé.
- `SwapTotal` — Quantité totale de mémoire swap disponible (exprimée en Ko).
- `SwapFree` — Quantité totale de mémoire swap libre (exprimée en Ko).

5.2.20. /proc/misc

Ce fichier affiche la liste des pilotes divers enregistrés sur le périphérique principal divers, portant le numéro 10:

```
135 rtc
  1 psaux
134 apm_bios
```

La première colonne correspond au nombre mineur de chaque périphérique et la deuxième indique le pilote utilisé.

5.2.21. /proc/modules

Ce fichier affiche une liste de tous les modules qui ont été chargés dans le noyau. Son contenu varie en fonction de la configuration et de l'utilisation du système, mais il devrait être organisé de façon semblable à la sortie du fichier exemple `/proc/modules` ci-dessous:

```
ide-cd                 27008   0 (autoclean)
cdrom                  28960   0 (autoclean) [ide-cd]
soundcore               4100   0 (autoclean)
agpgart                31072   0 (unused)
binfmt_misc             5956   1
iscsi                  32672   0 (unused)
scsi_mod               94424   1 [iscsi]
autofs                 10628   0 (autoclean) (unused)
tulip                  48608   1
ext3                   60352   2
jbd                    39192   2 [ext3]
```

La première colonne contient le nom du module. La deuxième indique la taille de la mémoire du module, en octets. La troisième indique si le module est actuellement chargé (`1`) ou non (`0`). La dernière colonne indique si le module peut se décharger automatiquement après une période d'inactivité (`autoclean`) ou s'il n'est pas utilisé (`unused`). Tout module ayant une ligne qui contient un nom entre parenthèses (`[` ou `]`) signifie que ce module dépend de la présence d'un autre module pour fonctionner.

5.2.22. /proc/mounts

Ce fichier fournit une brève liste de tous les montages utilisés par le système:

```
rootfs / rootfs rw 0 0
/dev/hda2 / ext3 rw 0 0
/proc /proc proc rw 0 0
/dev/hda1 /boot ext3 rw 0 0
none /dev/pts devpts rw 0 0
none /dev/shm tmpfs rw 0 0
none /proc/sys/fs/binfmt_misc binfmt_misc rw 0 0
```

Cette sortie est semblable au contenu de /etc/mtab, mis à part que /proc/mount est plus actuel.

La première colonne spécifie le périphérique monté et la deuxième indique le point de montage. La troisième colonne donne le type de système de fichiers et la quatrième vous indique s'il est monté en lecture-seule (ro) ou en lecture et écriture (rw). Les cinquième et sixième colonnes sont des valeurs fictives conçues pour correspondre au format utilisé dans /etc/mtab.

5.2.23. /proc/mtrr

Ce fichier fait référence aux MTRR (Memory Type Range Registers) utilisés avec le système. Si l'architecture de votre système prend en charge les MTRR, votre fichier /proc/mtrr pourrait avoir l'aspect suivant:

```
reg00: base=0x00000000 (   0MB), size=  64MB: write-back, count=1
```

Les MTRR sont utilisés avec les processeurs de la famille P6 d'Intel (Pentium II et supérieur) pour contrôer l'accès du processeur aux plages de mémoire. En utilisant une carte vidéo sur un bus PCI ou AGP, un fichier /proc/mtrr correctement configuré peut augmenter les performances de plus de 150%.

Dans la plupart des cas, cette valeur est correctement configurée pour vous. Pour avoir plus de renseignements sur les MTRR et la configuration manuelle de ce fichier, reportez-vous à l'adresse suivante: http://web1.linuxhq.com/kernel/v2.3/doc/mtrr.txt.html.

5.2.24. /proc/partitions

La plupart des informations présentées ici ne sont pas importantes pour l'utilisateur, à l'exception des colonnes suivantes:

- major — Le nombre majeur du périphérique avec cette partition. Le nombre majeur de notre exemple (3) correspond au périphérique bloc ide0 de /proc/devices.
- minor — Le nombre mineur du périphérique avec cette partition. Cela permet de séparer les partitions en différents périphériques physiques et fait référence au nombre situé à la fin du nom de la partition.
- #blocks — Répertorie le nombre de blocs de disque physique contenus dans une partition donnée.
- name — Nom de la partition.

5.2.25. /proc/pci

Ce fichier contient une liste complète des périphériques PCI de votre système. Selon le nombre de périphériques PCI présents sur votre système /proc/pci peut être assez long. Ci-après se trouve un exemple de ce fichier sur un système de base:

```
Bus  0, device   0, function  0:
  Host bridge: Intel Corporation 440BX/ZX - 82443BX/ZX Host bridge (rev 3).
    Master Capable.  Latency=64.
    Prefetchable 32 bit memory at 0xe4000000 [0xe7ffffff].
Bus  0, device   1, function  0:
  PCI bridge: Intel Corporation 440BX/ZX - 82443BX/ZX AGP bridge (rev 3).
    Master Capable.  Latency=64.  Min Gnt=128.
Bus  0, device   4, function  0:
  ISA bridge: Intel Corporation 82371AB PIIX4 ISA (rev 2).
Bus  0, device   4, function  1:
  IDE interface: Intel Corporation 82371AB PIIX4 IDE (rev 1).
    Master Capable.  Latency=32.
    I/O at 0xd800 [0xd80f].
Bus  0, device   4, function  2:
  USB Controller: Intel Corporation 82371AB PIIX4 USB (rev 1).
    IRQ 5.
    Master Capable.  Latency=32.
    I/O at 0xd400 [0xd41f].
Bus  0, device   4, function  3:
  Bridge: Intel Corporation 82371AB PIIX4 ACPI (rev 2).
    IRQ 9.
Bus  0, device   9, function  0:
  Ethernet controller: Lite-On Communications Inc LNE100TX (rev 33).
    IRQ 5.
    Master Capable.  Latency=32.
    I/O at 0xd000 [0xd0ff].
    Non-prefetchable 32 bit memory at 0xe3000000 [0xe30000ff].
Bus  0, device  12, function  0:
  VGA compatible controller: S3 Inc. ViRGE/DX or /GX (rev 1).
    IRQ 11.
    Master Capable.  Latency=32.  Min Gnt=4.Max Lat=255.
    Non-prefetchable 32 bit memory at 0xdc000000 [0xdfffffff].
```

Cette sortie affiche une liste de tous les périphériques PCI, triés par ordre de bus, périphérique et fonction. En plus de fournir le nom et la version du périphérique, cette liste vous donne des informations IRQ détaillées afin que vous puissiez détecter rapidement les conflits.



Astuce

Pour obtenir une version plus lisible de ce genre d'informations, tapez:

```
/sbin/lspci -vb
```

5.2.26. /proc/slabinfo

Ce fichier fournit des informations sur l'utilisation de la mémoire au niveau bloc (*slab*). Les noyaux Linux supérieurs à 2.2 utilisent des *groupes d'emplacement mémoire de type bloc* pour gérer la mémoire au-dessus du niveau page. Les objets couramment utilisés ont leurs propres groupes d'emplacement mémoire de type bloc. Ci-dessous figure une partie d'un fichier virtuel `/proc/slabinfo` typique:

```
slabinfo - version: 1.1
kmem_cache             64     68    112     2     2     1
nfs_write_data          0      0    384     0     0     1
nfs_read_data           0    160    384     0    16     1
nfs_page                0    200     96     0     5     1
ip_fib_hash            10    113     32     1     1     1
journal_head           51   7020     48     2    90     1
revoke_table            2    253     12     1     1     1
revoke_record           0      0     32     0     0     1
clip_arp_cache          0      0    128     0     0     1
ip_mrt_cache            0      0     96     0     0     1
```

Les valeurs de ce fichier sont présentées selon l'ordre suivant: nom du cache, nombre d'objets actifs, nombre total d'objets, taille des objets, nombre de blocs (slabs) actifs des objets, nombre total de blocs des objets et nombre de pages par bloc.

Veuillez noter que *actif* signifie ici qu'un objet est en cours d'utilisation.

5.2.27. /proc/stat

Ce fichier effectue le suivi de différentes statistiques sur le système depuis le dernier redémarrage. Le contenu de `/proc/stat`, qui peut être plutôt long, commence de la façon suivante:

```
cpu  1139111 3689 234449 84378914
cpu0 1139111 3689 234449 84378914
page 2675248 8567956
swap 10022 19226
intr 93326523 85756163 174412 0 3 3 0 6 0 1 0 428620 0 60330 0 1368304 5538681
disk_io: (3,0):(1408049,445601,5349480,962448,17135856)
ctxt 27269477
btime 886490134
processes 206458
```

Ci-après se trouve une liste des statistiques les plus utilisées:

- `cpu` — Mesure le nombre de *jiffies* (1/100 de seconde) pendant lequel le système a été respectivement en mode utilisateur, en mode utilisateur avec basse priorité (nice), en mode système et au repos. Le total pour chacune des unités centrales est indiqué au sommet et chaque unité centrale individuelle est répertoriée en dessous avec ses propres statistiques.
- `page` — Nombre de pages mémoire que le système a enregistrées en entrée et en sortie.
- `swap` — Nombre de pages échangées par le système.
- `intr` — Nombre d'interruptions reçues par le système.
- `btime` — Temps du démarrage, mesuré en nombre de secondes écoulées depuis le 1er janvier 1970; ce que l'on appelle aussi parfois l'*époque*.

5.2.28. /proc/swaps

Ce fichier mesure l'espace swap et son utilisation. Pour un système n'ayant qu'une seule partition swap, la sortie de `/proc/swap` peut ressembler à l'extrait ci-dessous:

```
Filename      Type        Size    Used   Priority
/dev/hda6     partition  136512  20024  -1
```

Bien qu'il soit possible de trouver une partie de ces informations dans d'autres fichiers du répertoire `/proc/`, `/proc/swap` fournit un instantané de chaque nom de fichier swap, du type d'espace swap et de la taille totale et de l'espace utilisé (exprimée en Ko). La colonne "priority" est utile lorsque plusieurs fichiers swap sont en cours d'utilisation. Plus la priorité est basse, plus il est possible que le fichier swap soit utilisé.

5.2.29. /proc/uptime

Ce fichier contient des informations sur le temps de fonctionnement du système depuis le dernier redémarrage. La sortie de `/proc/uptime` est succincte:

```
350735.47 234388.90
```

Le premier nombre vous indique le nombre total de secondes de fonctionnement depuis le démarrage. Le second vous indique, en secondes également, la période d'inactivité de l'ordinateur pendant ce même temps.

5.2.30. /proc/version

Ce fichier vous indique les versions du noyau Linux et de `gcc` utilisées, de même que la version de Red Hat Linux installée sur le système:

```
Linux version 2.4.20-0.40 (user@foo.redhat.com)(gcc version 3.2.1 20021125
(Red Hat Linux 8.0 3.2.1-1)) #1 Tue Dec 3 20:50:18 EST 2002
```

Ces informations ont diverses fonctions, telles que de fournir des données sur la version lorsqu'un utilisateur se connecte.

5.3. Répertoires de /proc/

Les groupes communs d'informations sur le noyau sont regroupés en répertoires et sous-répertoires dans `/proc/`.

5.3.1. Répertoires de processus

Chaque répertoire `/proc/` contient un certain nombre de répertoires nommés à partir de chiffres. Ci-dessous figure un exemple de début de listing:

```
dr-xr-xr-x    3 root     root            0 Feb 13 01:28 1
dr-xr-xr-x    3 root     root            0 Feb 13 01:28 1010
dr-xr-xr-x    3 xfs      xfs             0 Feb 13 01:28 1087
dr-xr-xr-x    3 daemon   daemon          0 Feb 13 01:28 1123
dr-xr-xr-x    3 root     root            0 Feb 13 01:28 11307
dr-xr-xr-x    3 apache   apache          0 Feb 13 01:28 13660
dr-xr-xr-x    3 rpc      rpc             0 Feb 13 01:28 637
```

```
dr-xr-xr-x    3 rpcuser  rpcuser         0 Feb 13 01:28 666
```

Ces répertoires sont appelés *répertoires de processus* car ils font référence à un ID de processus et contiennent des informations se rapportant à ce processus. Le propriétaire et le groupe de chaque répertoire de processus est paramétré sur l'utilisateur qui exécute le processus. Lorsque le processus est terminé, son répertoire de processus `/proc/` disparaît.

Chaque répertoire de processus contient les lignes suivantes:

- `cmdline` — Ce fichier contient la commande émise au début du processus.
- `cpu` — Fournit des informations spécifiques sur l'utilisation de chaque unité centrale du système. Un processus exécuté sur un système à double unité centrale produit une sortie semblable à l'extrait ci-dessous:
  ```
  cpu  11 3
  cpu0 0 0
  cpu1 11 3
  ```
- `cwd` — Lien symbolique vers le répertoire de travail courant pour ce processus.
- `environ` — Fournit la liste des variables d'environnement du processus. La variable d'environnement est indiquée en majuscules et la valeur en minuscules.
- `exe` — Lien symbolique vers le fichier exécutable de ce processus.
- `fd` — Répertoire qui contient tous les descripteurs de fichiers pour un processus donné. Ces derniers sont fournis en liens numérotés:
  ```
  total 0
  lrwx------    1 root     root           64 May  8 11:31 0 -> /dev/null
  lrwx------    1 root     root           64 May  8 11:31 1 -> /dev/null
  lrwx------    1 root     root           64 May  8 11:31 2 -> /dev/null
  lrwx------    1 root     root           64 May  8 11:31 3 -> /dev/ptmx
  lrwx------    1 root     root           64 May  8 11:31 4 -> socket:[7774817]
  lrwx------    1 root     root           64 May  8 11:31 5 -> /dev/ptmx
  lrwx------    1 root     root           64 May  8 11:31 6 -> socket:[7774829]
  lrwx------    1 root     root           64 May  8 11:31 7 -> /dev/ptmx
  ```
- `maps` — Contient les configurations mémoire vers les divers fichiers exécutables et les bibliothèques associés à ce processus. Ce fichier peut être long, en fonction de la complexité du processus. Ci-après figure un exemple de début de sortie du processus `sshd`:
  ```
  08048000-08086000 r-xp 00000000 03:03 391479     /usr/sbin/sshd
  08086000-08088000 rw-p 0003e000 03:03 391479     /usr/sbin/sshd
  08088000-08095000 rwxp 00000000 00:00 0
  40000000-40013000 r-xp 00000000 03:03 293205     /lib/ld-2.2.5.so
  40013000-40014000 rw-p 00013000 03:03 293205     /lib/ld-2.2.5.so
  40031000-40038000 r-xp 00000000 03:03 293282     /lib/libpam.so.0.75
  40038000-40039000 rw-p 00006000 03:03 293282     /lib/libpam.so.0.75
  40039000-4003a000 rw-p 00000000 00:00 0
  4003a000-4003c000 r-xp 00000000 03:03 293218     /lib/libdl-2.2.5.so
  4003c000-4003d000 rw-p 00001000 03:03 293218     /lib/libdl-2.2.5.so
  ```
- `mem` — Mémoire retenue par le processus. Ce fichier ne peut être lu par l'utilisateur.
- `root` — Lien vers le répertoire root du processus.
- `stat` — État du processus.
- `statm` — État de la mémoire utilisée par le processus. Ci-dessous figure un exemple de fichier `/proc/statm`:
  ```
  263 210 210 5 0 205 0
  ```
  Les sept colonnes font référence à différentes statistiques de mémoire pour le processus. De gauche à droite, elles indiquent les aspects suivants de la mémoire utilisée:

1. Taille totale du programme (exprimée en Ko);
2. Taille des portions de mémoire (exprimée en Ko);
3. Nombre de pages partagées;
4. Nombre de pages de code;
5. Nombre de pages de données/pile;
6. Nombre de pages de bibliothèque;
7. Nombre de pages incorrectes.

- `status` — État du processus sous une forme plus lisible que `stat` ou `statm`. Un exemple de sortie de `sshd` ressemble à l'extrait ci-dessous:

```
Name: sshd
State: S (sleeping)
Tgid: 797
Pid: 797
PPid: 1
TracerPid: 0
Uid: 0 0 0 0
Gid: 0 0 0 0
FDSize: 32
Groups:
VmSize:     3072 kB
VmLck:         0 kB
VmRSS:       840 kB
VmData:      104 kB
VmStk:        12 kB
VmExe:       300 kB
VmLib:      2528 kB
SigPnd: 0000000000000000
SigBlk: 0000000000000000
SigIgn: 8000000000001000
SigCgt: 0000000000014005
CapInh: 0000000000000000
CapPrm: 00000000fffffeff
CapEff: 00000000fffffeff
```

  Les informations contenues dans cette sortie incluent le nom et l'ID du processus, l'état (tel que `S (sleeping)` pour le mode veille, ou `R (running)`) pour une exécution en cours, l'ID de l'utilisateur/du groupe qui exécute le processus de même que des données beaucoup plus détaillées portant sur l'utilisation de la mémoire.

5.3.1.1. /proc/self/

Le répertoire `/proc/self/` est un lien vers le processus en cours d'exécution. Cela permet à un processus de se contrôler lui-même sans avoir à connaître son ID de processus.

Dans un environnement shell, le résultat est le même, que vous répertoriiez le contenu du répertoire `/proc/self/` ou celui du répertoire de processus pour ce processus.

5.3.2. /proc/bus/

Ce répertoire contient des informations spécifiques aux divers bus disponibles sur le système. Ainsi, par exemple, sur un système standard comportant des bus ISA, PCI et USB, les informations actuelles relatives à chacun de ces bus se trouvent dans son répertoire sous /proc/bus/.

Le contenu des sous-répertoires et des fichiers disponibles diffère grandement selon la configuration précise de votre système. Cependant, chaque répertoire pour chacun des types de bus contient au moins un répertoire pour chaque bus de ce type. Ces répertoires individuels de bus, généralement indiqués par des chiffres, tels que 00, contiennent des fichiers binaires qui font référence aux divers périphériques disponibles sur ce bus.

Par exemple, un système ayant un bus USB auquel aucun périphérique n'est connecté, a un répertoire /proc/bus/usb/ qui contient plusieurs fichiers:

```
total 0
dr-xr-xr-x    1 root     root            0 May  3 16:25 001
-r--r--r--    1 root     root            0 May  3 16:25 devices
-r--r--r--    1 root     root            0 May  3 16:25 drivers
```

Le répertoire /proc/bus/usb/ contient des fichiers qui détectent les différents périphériques sur les bus USB, ainsi que les pilotes nécessaires pour les utiliser. Le répertoire /proc/bus/usb/001/ contient tous les périphériques présents sur le premier bus USB. En examinant le contenu du fichier devices vous pouvez identifier le concentrateur root USB sur la carte mère:

```
T:  Bus=01 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#=  1 Spd=12  MxCh= 2
B:  Alloc=  0/900 us ( 0%), #Int=  0, #Iso=  0
D:  Ver= 1.00 Cls=09(hub  ) Sub=00 Prot=00 MxPS= 8 #Cfgs=  1
P:  Vendor=0000 ProdID=0000 Rev= 0.00
S:  Product=USB UHCI Root Hub
S:  SerialNumber=d400
C:* #Ifs= 1 Cfg#= 1 Atr=40 MxPwr=  0mA
I:  If#= 0 Alt= 0 #EPs= 1 Cls=09(hub  ) Sub=00 Prot=00 Driver=hub
E:  Ad=81(I) Atr=03(Int.) MxPS=   8 Ivl=255ms
```

5.3.3. /proc/driver/

Ce répertoire contient des informations sur des pilotes spécifiques utilisés par le noyau.

On peut y trouver un fichier commun, rtc, qui fournit une sortie provenant du pilote pour l'*horloge temps réel (RTC)* (de l'anglais 'Real Time Clock') du système, le dispositif qui maintient l'heure lorsque le système est éteint. Ci-après figure un exemple de sortie de /proc/driver/rtc:

```
rtc_time : 01:38:43
rtc_date : 1998-02-13
rtc_epoch : 1900
alarm  : 00:00:00
DST_enable : no
BCD  : yes
24hr  : yes
square_wave : no
alarm_IRQ : no
update_IRQ : no
periodic_IRQ : no
periodic_freq : 1024
batt_status : okay
```

Pour plus d'informations sur l'horloge temps réel (RTC), consultez /usr/src/linux-2.4/Documentation/rtc.txt.

5.3.4. /proc/fs

Ce répertoire montre quels fichiers système sont exportés. Si vous exécutez un serveur NFS, vous pouvez taper cat /proc/fs/nfs/exports afin d'afficher les systèmes de fichiers qui sont partagés ainsi que les autorisations accordées pour ces derniers. Pour plus d'informations sur le partage des fichiers système avec NFS, consultez le Chapitre 9.

5.3.5. /proc/ide/

Ce répertoire contient des informations sur les périphériques IDE du système. Chaque canal IDE est représenté par un répertoire séparé, tel que /proc/ide/ide0 et /proc/ide/ide1. De plus, un fichier drivers est aussi disponible; il fournit le numéro de version des divers pilotes utilisés sur les canaux IDE:

```
ide-cdrom version 4.59
ide-floppy version 0.97
ide-disk version 1.10
```

Plusieurs jeux de puces ou chipsets fournissent également dans ce répertoire un fichier d'informations qui donne des renseignements supplémentaires sur lecteurs connectés via les canaux. Par exemple, un chipset générique Ultra 33 PIIX4 d'Intel produit un fichier /proc/ide/piix qui vous indiquera si DMA ou UDMA est activé pour les périphériques situés sur les canaux IDE:

```
                              Intel PIIX4 Ultra 33 Chipset.
------------- Primary Channel ---------------- Secondary Channel -------------

                enabled                           enabled
------------- drive0 --------- drive1 -------- drive0 ---------- drive1 ------

DMA enabled:    yes              no              yes                no
UDMA enabled:   yes              no              no                 no
UDMA enabled:   2                X               X                  X
UDMA
DMA
PIO
```

En examinant le répertoire d'un canal IDE, tel que ide0, vous pouvez obtenir des informations supplémentaires. Le fichier channel indique le numéro de canal, alors que model vous indique le type de bus (tel que pci).

5.3.5.1. Le répertoire de périphérique

À l'intérieur de chaque répertoire de canal IDE se trouve un répertoire de périphérique. Le nom du répertoire de périphérique correspond à la lettre du périphérique dans le répertoire /dev/. Par exemple, le premier périphérique IDE sur ide0 serait hda.



Remarque

Il existe un lien symbolique pour chacun de ces répertoires de périphériques dans le répertoire /proc/ide/.

Chaque répertoire de périphérique contient un recueil d'informations et de statistiques. Le contenu de ces répertoires varient selon le type de périphérique connecté. Parmi les fichiers les plus utiles communs à beaucoup de périphériques se trouvent:

- cache — le cache du périphérique;
- capacity — la capacité du périphérique, en blocs de 512 octets;
- driver — le pilote et la version utilisés pour contrôler le périphérique;
- geometry — la géométrie physique et logique du périphérique;
- media — le type de périphérique, comme par exemple disk.
- model — le nom ou le numéro de modèle du périphérique;
- settings — un ensemble de paramètres courants du périphérique. Ce fichier contient normalement un certain nombre d'informations techniques utiles. Un exemple de fichier settings pour un disque dur IDE standard ressemble à l'extrait ci-dessous:

```
name                value          min          max          mode
----                -----          ---          ---          ----
bios_cyl            784            0            65535        rw
bios_head           255            0            255          rw
bios_sect           63             0            63           rw
breada_readahead    4              0            127          rw
bswap               0              0            1            r
current_speed       66             0            69           rw
file_readahead      0              0            2097151      rw
ide_scsi            0              0            1            rw
init_speed          66             0            69           rw
io_32bit            0              0            3            rw
keepsettings        0              0            1            rw
lun                 0              0            7            rw
max_kb_per_request  64             1            127          rw
multcount           8              0            8            rw
nice1               1              0            1            rw
nowerr              0              0            1            rw
number              0              0            3            rw
pio_mode            write-only     0            255          w
slow                0              0            1            rw
unmaskirq           0              0            1            rw
using_dma           1              0            1            rw
```

5.3.6. /proc/irq/

Ce répertoire est utilisé pour paramétrer l'association IRQ-CPU, qui permet de connecter un IRQ donné à une seule unité centrale. Vous pouvez également empêcher qu'une unité centrale gère un IRQ.

Chaque IRQ a son propre répertoire, ce qui permet une configuration individuelle de chacun d'eux. Le fichier /proc/irq/prof_cpu_mask est un masque de bit qui contient les valeurs par défaut pour le fichier smp_affinity dans le répertoire IRQ. Les valeurs de smp_affinity spécifient quelles unités centrales gèrent cet IRQ spécifique.

Pour obtenir davantage d'informations sur le répertoire /proc/irq/ consultez:

```
/usr/src/linux-2.4/Documentation/filesystems/proc.txt
```

5.3.7. /proc/net/

Ce répertoire fournit une vision exhaustive de nombreux paramètres et statistiques réseau. Chaque fichier couvre une gamme spécifique d'informations relatives à la gestion du réseau sur le système. Vous trouverez ci-dessous une liste partielle de ces fichiers virtuels:

- arp — Contient la table ARP du noyau. Ce fichier est particulièrement utile pour connecter une adresse câblée à une adresse IP sur un système.
- atm — Un répertoire contenant des fichiers avec divers paramètres et statistiques de *mode de transfert asynchrone (ATM)* (de l'anglais, 'Asynchronous Transfer Mode'). Ce répertoire est principalement utilisé pour la gestion de réseau ATM et les cartes ADSL.
- dev — Répertorie les différents périphériques réseau configurés sur le système, complet avec les statistiques de transmission et de réception. Ce fichier vous indique rapidement le nombre d'octets envoyés et reçus par chaque interface, le nombre de paquets entrants et sortants, le nombre d'erreurs trouvées, le nombre de paquets perdus, etc.
- dev_mcast — Affiche les différents groupes de multidiffusion Layer2 qu'écoute chaque périphérique.
- igmp — Affiche la liste des adresses IP de multidiffusion auxquelles le système s'est joint.
- ip_fwchains — Si les ipchains sont en cours d'utilisation, ce fichier virtuel indique toutes les règles actuelles.
- ip_fwnames — Si les ipchains sont en cours d'utilisation, ce fichier virtuel répertorie tous les noms de chaînes de pare-feu.
- ip_masquerade — Fournit une table d'informations relatives aux usurpations d'identité sous ipchains.
- ip_mr_cache — Liste du cache du routeur de diffusion.
- ip_mr_vif — Liste des interfaces virtuelles de diffusion.
- netstat — Contient un ensemble large, mais détaillé, de statistiques réseau, telles que les délais d'attente TCP, les cookies SYN envoyés et reçus, etc.
- psched — Liste des paramètres du programmateur global des paquets.
- raw — Liste des statistiques brutes relatives aux périphériques.
- route — Affiche la table de routage du noyau.
- rt_cache — Contient le cache de routage actuel.
- snmp — Liste des données du protocole d'administration à distance de réseaux ou SNMP (de l'anglais, 'Simple Network Management Protocol') pour divers protocoles de gestion de réseau en cours d'utilisation.
- sockstat — Fournit des statistiques sur les sockets.
- tcp — Contient des informations détaillées sur les sockets TCP.
- tr_rif — La table de routage RIF du bus annulaire à jeton (token ring).
- udp — Contient des informations détaillées sur les sockets UDP.
- unix — Liste les sockets de domaine UNIX actuellement utilisés.

- `wireless` — Répertorie les données d'interface sans fil.

5.3.8. /proc/scsi/

Ce répertoire est analogue au répertoire `/proc/ide/` à la seule différence près qu'il est réservé aux périphériques SCSI.

Le fichier principal est `/proc/scsi/scsi`, qui contient une liste de tous les périphériques SCSI reconnus. Cette liste fournit également des informations sur le type de périphérique, ainsi que le nom de modèle, le fabricant, le canal et les données ID SCSI disponibles.

Par exemple, si un système disposait d'un lecteur de CD-ROM, d'un lecteur de bande, de disques durs ainsi que d'un contrôleur RAID, ce fichier ressemblerait à l'extrait ci-dessous:

```
Attached devices:
Host: scsi1 Channel: 00 Id: 05 Lun: 00
  Vendor: NEC      Model: CD-ROM DRIVE:466 Rev: 1.06
  Type:   CD-ROM                           ANSI SCSI revision: 02
Host: scsi1 Channel: 00 Id: 06 Lun: 00
  Vendor: ARCHIVE  Model: Python 04106-XXX Rev: 7350
  Type:   Sequential-Access                ANSI SCSI revision: 02
Host: scsi2 Channel: 00 Id: 06 Lun: 00
  Vendor: DELL     Model: 1x6 U2W SCSI BP  Rev: 5.35
  Type:   Processor                        ANSI SCSI revision: 02
Host: scsi2 Channel: 02 Id: 00 Lun: 00
  Vendor: MegaRAID Model: LD0 RAID5 34556R Rev: 1.01
  Type:   Direct-Access                    ANSI SCSI revision: 02
```

Chaque pilote SCSI utilisé par le système a son propre répertoire dans `/proc/scsi/`, qui contient des fichiers spécifiques à chaque contrôleur SCSI qui utilise ce pilote. Par conséquent, dans le cas de l'exemple ci-dessus, les répertoires `aic7xxx` et `megaraid` sont présents, car ces deux pilotes sont utilisés. Les fichiers situés dans chacun des répertoires contiennent généralement la plage d'adresses E/S, les IRQ ainsi que les statistiques relatives au contrôleur SCSI qui utilise ce pilote. Chaque contrôleur peut rapporter différents types et quantités d'informations. Le fichier du contrôleur SCSI Adaptec AIC-7880 Ultra produit dans cet exemple la sortie suivante:

```
Adaptec AIC7xxx driver version: 5.1.20/3.2.4
Compile Options:
  TCQ Enabled By Default : Disabled
  AIC7XXX_PROC_STATS     : Enabled
  AIC7XXX_RESET_DELAY    : 5

Adapter Configuration:
           SCSI Adapter: Adaptec AIC-7880 Ultra SCSI host adapter
                           Ultra Narrow Controller
    PCI MMAPed I/O Base: 0xfcffe000
 Adapter SEEPROM Config: SEEPROM found and used.
      Adaptec SCSI BIOS: Enabled
                    IRQ: 30
                   SCBs: Active 0, Max Active 1,
                         Allocated 15, HW 16, Page 255
             Interrupts: 33726
      BIOS Control Word: 0x18a6
   Adapter Control Word: 0x1c5f
   Extended Translation: Enabled
Disconnect Enable Flags: 0x00ff
     Ultra Enable Flags: 0x0020
 Tag Queue Enable Flags: 0x0000
```

```
Ordered Queue Tag Flags: 0x0000
Default Tag Queue Depth: 8
    Tagged Queue By Device array for aic7xxx host instance 1:
      {255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255}
    Actual queue depth per device for aic7xxx host instance 1:
      {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1}

Statistics:


(scsi1:0:5:0)
  Device using Narrow/Sync transfers at 20.0 MByte/sec, offset 15
  Transinfo settings: current(12/15/0/0), goal(12/15/0/0), user(12/15/0/0)
  Total transfers 0 (0 reads and 0 writes)
             < 2K      2K+     4K+     8K+    16K+    32K+    64K+   128K+
   Reads:       0        0       0       0       0       0       0       0
  Writes:       0        0       0       0       0       0       0       0


(scsi1:0:6:0)
  Device using Narrow/Sync transfers at 10.0 MByte/sec, offset 15
  Transinfo settings: current(25/15/0/0), goal(12/15/0/0), user(12/15/0/0)
  Total transfers 132 (0 reads and 132 writes)
             < 2K      2K+     4K+     8K+    16K+    32K+    64K+   128K+
   Reads:       0        0       0       0       0       0       0       0
  Writes:       0        0       0       1     131       0       0       0
```

Cet écran vous permet de visualiser la vitesse de transfert des différents périphériques SCSI connectés au contrôleur en fonction de l'ID de canal, ainsi que des statistiques détaillées concernant la quantité et la taille des fichiers lus ou écrits par ces périphériques. Par exemple, à partir de la sortie ci-dessus, vous pouvez voir que ce contrôleur communique avec le lecteur de CD-ROM à une vitesse de 20 Mo par seconde, alors que le lecteur de bande n'est connecté lui qu'à 10 Mo par seconde.

5.3.9. /proc/sys/

Le répertoire `/proc/sys/` est différent des autres répertoires de `/proc/` car, en plus de fournir des informations relatives au système, il vous permet d'apporter des modifications à la configuration du noyau. Ceci permet à l'administrateur de l'ordinateur d'activer et de désactiver immédiatement des fonctions du noyau.



Soyez prudent lorsque vous modifiez les paramètres sur un système de production à l'aide des différents fichiers du répertoire `/proc/sys/`. La modification d'un mauvais paramètre peut rendre le noyau instable et nécessiter le redémarrage du système.

Pour cette raison, avant de changer une valeur dans `/proc/sys/`, assurez-vous que les options de ce fichier sont bien valides.

Pour savoir si un fichier donné peut être configuré ou s'il est uniquement conçu pour fournir des informations, vous pouvez l'afficher dans le terminal à l'aide de l'option `-l` entrée à l'invite du shell. option at the shell prompt. Si le fichier peut être modifié, vous pouvez alors l'utiliser pour configurer le noyau. Ci-dessous figure un exemple d'affichage partiel de `/proc/sys/fs`:

```
-r--r--r--    1 root     root            0 May 10 16:14 dentry-state
```

```
-rw-r--r--    1 root     root            0 May 10 16:14 dir-notify-enable
-r--r--r--    1 root     root            0 May 10 16:14 dquot-nr
-rw-r--r--    1 root     root            0 May 10 16:14 file-max
-r--r--r--    1 root     root            0 May 10 16:14 file-nr
```

Dans cet exemple, les fichiers `dir-notify-enable` et `file-max` peuvent être modifiés et, par conséquent, peuvent être utilisés pour configurer le noyau. Les autres fichiers ne fournissent que des informations sur les paramètres actuels.

Pour changer une valeur dans un fichier `/proc/sys/`, il faut enregistrer la nouvelle valeur dans le fichier à l'aide de la commande echo. Par exemple, pour activer la touche d'interrogation système sur un noyau en cours d'exécution, tapez la commande :

```
echo 1 > /proc/sys/kernel/sysrq
```

Cette opération aura pour effet de modifier la valeur `sysrq` du fichier, qui passera de `0` (off) à `1` (on).

La touche d'interrogation système a été conçue pour vous permettre d'indiquer au noyau d'exécuter un certain nombre d'opérations importantes au moyen d'une simple combinaison de touches, comme par exemple arrêter ou redémarrer immédiatement un système, synchroniser tous les systèmes de fichiers montés ou vider des informations importantes sur votre console. Cette fonction est particulièrement utile lorsque vous utilisez un noyau de développement ou si vous rencontrez des blocages de système. Elle est toutefois considérée comme un risque de sécurité pour une console automatique et est donc désactivée par défaut sous Red Hat Linux.

Reportez-vous à `/usr/src/linux-2.4/Documentation/sysrq.txt` afin d'obtenir davantage d'informations sur la touche d'interrogation système.

Quelques fichiers de configuration `/proc/sys/` contiennent plus d'une valeur. Placez un espace blanc entre chaque valeur transmise à l'aide de la commande `echo` afin d'envoyer correctement les nouvelles valeurs, comme c'est le cas l'exemple suivant:

```
echo 4 2 45 > /proc/sys/kernel/acct
```



Remarque

Toute modification de configuration effectuée à l'aide de la commande `echo` disparaîtra lorsque le système est redémarré. Pour faire en sorte que vos modifications soient appliquées au démarrage, reportez-vous à la Section 5.4.

Le répertoire `/proc/sys/` contient plusieurs sous-répertoires qui contrôlent différents aspects d'un noyau en cours d'exécution.

5.3.9.1. `/proc/sys/dev/`

Ce répertoire fournit des paramètres pour des périphériques particuliers du système. La plupart des systèmes ont au moins deux répertoires, à savoir `cdrom` et `raid`. Les noyaux personnalisés eux peuvent en avoir d'autres, tels que `parport`, qui donne la possibilité de partager un port parallèle entre plusieurs pilotes de périphériques.

Le répertoire `cdrom` contient un fichier appelé `info`, qui indique un certain nombre de paramètres importants pour le lecteur de CD-ROM:

```
CD-ROM information, Id: cdrom.c 3.12 2000/10/18

drive name:  hdc
```

```
drive speed:  32
drive # of slots: 1
Can close tray:  1
Can open tray:  1
Can lock tray:  1
Can change speed: 1
Can select disk: 0
Can read multisession: 1
Can read MCN:  1
Reports media changed: 1
Can play audio:  1
Can write CD-R:  0
Can write CD-RW: 0
Can read DVD:  0
Can write DVD-R: 0
Can write DVD-RAM: 0
```

Ce fichier peut être examiné rapidement pour découvrir les qualités d'un lecteur de CD-ROM inconnu, pour le noyau tout au moins. Si plusieurs lecteurs de CD-ROM sont disponibles sur un système, chaque périphérique dispose de sa propre colonne d'informations.

De nombreux fichiers de `/proc/sys/dev/cdrom`, tels que `autoclose` et `checkmedia`, peuvent être utilisés pour contrôler le lecteur de CD-ROM du système. Utilisez simplement la commande `echo` pour activer ou désactiver ces fonctions.

Si la prise en charge de RAID est compilée dans le noyau, un répertoire `/proc/sys/dev/raid/` sera disponible et contiendra au moins deux fichiers: `speed_limit_min` et `speed_limit_max`. Ces paramètres permettent de déterminer quelle augmentation de vitesse appliquer au périphérique RAID pour des tâches E/S particulièrement intensives, telles que la re-synchronisation des disques.

5.3.9.2. `/proc/sys/fs/`

Ce répertoire contient une gamme d'options et d'informations relatives à divers aspects des systèmes de fichiers, y compris, quota, indicateur de fichier, inode et dentry.

Le répertoire `binfmt_misc` est utilisé pour fournir au noyau la prise en charge de formats binaires divers.

Les fichiers importants du répertoire `/proc/sys/fs` comprennent:

- `dentry-state` — donne l'état du cache du répertoire. Le fichier ressemble à l'extrait ci-dessous:
  `57411 52939 45 0 0 0`

  Le premier nombre indique le nombre total d'entrées dans le cache du répertoire, alors que le deuxième indique le nombre d'entrées non utilisées. Le troisième indique le nombre de secondes entre le moment où un répertoire a été libéré et le moment où il peut être récupéré et le quatrième mesure les pages actuellement demandées par le système. Les deux derniers nombres ne sont pas utilisés et n'affichent actuellement que des zéros.
- `dquot-nr` — indique le nombre maximum d'entrées de quota de disque en cache.
- `file-max` — permet de changer le nombre maximum d'indicateurs de fichier alloués par le noyau. Si vous augmentez la valeur dans ce fichier, vous pourrez résoudre des erreurs causées par le manque d'indicateurs de fichier disponibles.
- `file-nr` — affiche le nombre d'indicateurs de fichier alloués, utilisés et maximum.
- `overflowgid` et `overflowuid` — définissent respectivement l'ID groupe et l'ID utilisateur fixes; ils sont utilisés avec des systèmes de fichiers qui ne prennent en charge que des ID groupe et utilisateur 16 bits.

- super-max — contrôle le nombre maximum de superblocs disponibles.
- super-nr — affiche le nombre actuel de superblocs utilisés.

5.3.9.3. /proc/sys/kernel/

Ce répertoire contient divers fichiers de configuration qui affectent directement le fonctionnement du noyau. Parmi les fichiers les plus importants figurent:

- acct — contrôle la suspension de la comptabilisation du processus sur la base du pourcentage d'espace libre disponible sur le système de fichiers contenant le journal. Par défaut, ce fichier ressemble à l'extrait ci-dessous:

  ```
  4 2 30
  ```

  La deuxième valeur définit le seuil de suspension de la journalisation en pourcentage d'espace libre, alors que la première valeur indique le pourcentage nécessaire pour reprendre la journalisation. La troisième valeur indique l'intervalle en secondes entre les interrogations du système de fichiers par le noyau pour savoir si la journalisation doit être suspendue ou reprise.

- cap-bound — contrôle les paramètres de *délimitation des capacités* qui fournit la liste des capacités de tout processus du système. Si une capacité n'est pas incluse dans cette liste, aucun processus, quels que soient ses privilèges, ne peut l'exécuter. L'objectif est d'améliorer la sécurité du système en s'assurant que certaines choses ne peuvent se produire, du moins au-delà d'un point donné du processus de démarrage.

  Pour obtenir une liste des valeurs acceptables pour ce fichier virtuel, consultez /usr/src/linux-2.4/include/linux/capability.h. De plus amples informations sur la délimitation des capacités sont disponibles en ligne à l'adresse suivante: http://lwn.net/1999/1202/kernel.php3.

- ctrl-alt-del — Contrôle si [Ctrl]-[Alt]-[Supprimer] redémarre correctement l'ordinateur à l'aide d'init (valeur 0) ou force un redémarrage immédiat sans synchroniser les tampons erronés vers le disque (valeur 1).
- domainname — permet de configurer le nom de domaine du système, tel que example.com.
- hostname — permet de configurer le nom d'hôte du système, tel que www.example.com.
- hotplug — configure l'utilitaire à utiliser lorsqu'un changement de configuration est détecté par le système. Il est surtout utilisé avec USB et Cardbus PCI. La valeur par défaut de /sbin/hotplug ne devrait pas être modifiée, à moins que vous ne testiez un nouveau programme pour jouer ce rôle.
- modprobe — définit l'emplacement du programme à utiliser pour charger des modules du noyau lorsque cela s'avère nécessaire. La valeur par défaut de /sbin/modprobe signifie que kmod l'appelle pour charger un module lorsqu'une unité d'exécution du noyau appelle kmod.
- msgmax — définit la taille maximum de tout message envoyé d'un processus à un autre; sa valeur par défaut est 8192 octets. Soyez prudent lorsque vous décidez d'augmenter cette valeur car les messages mis en file d'attente entre les processus sont stockés dans la mémoire non échangeable du noyau; Toute augmentation de msgmax augmentera également la demande de mémoire vive du système.
- msgmnb — définit le nombre maximum d'octets dans une file d'attente de messages. La valeur par défaut est 16384.
- msgmni — définit le nombre maximum d'identificateurs de file d'attente de messages. Par défaut, la valeur est 16.
- osrelease — fournit le numéro de version du noyau Linux. Ce fichier ne peut être modifié qu'en changeant la source du noyau et en recompilant.
- ostype — affiche le type de système d'exploitation. Par défaut, ce fichier est paramétré sur Linux; cette valeur ne peut être modifiée qu'en changeant la source du noyau et en recompilant.

- overflowgid et overflowuid — définissent respectivement l'ID groupe et l'ID utilisateur fixes; ils sont utilisés avec des appels système sur des architectures qui ne prennent en charge que des ID groupe et utilisateur 16 bits.
- panic — définit le nombre de secondes de report du redémarrage par le noyau, lorsque le système subit une panique du noyau. Par défaut, la valeur est de 0, ce qui désactive le redémarrage automatique après une panique.
- printk — ce fichier contrôle toute une série de paramètres relatifs à l'affichage ou à la journalisation de messages d'erreur. Chaque message d'erreur rapporté par le noyau a un *niveau journal* (loglevel) qui lui est associé et qui définit son importance. Les valeurs du niveau journal se répartissent dans l'ordre suivant:
  - 0 — Urgence du noyau. Le système est inutilisable.
  - 1 — Alerte du noyau. Une action immédiate est requise.
  - 2 — Condition du noyau considérée comme critique.
  - 3 — Condition générale d'erreur du noyau.
  - 4 — Condition générale d'avertissement du noyau.
  - 5 — Avis du noyau d'une condition normale, mais importante.
  - 6 — Message d'information du noyau.
  - 7 — Messages de niveau débogage du noyau.

  Le fichier printk comporte quatre valeurs:

  ```
  6 4 1 7
  ```

  Chacune de ces valeurs définit une règle différente de traitement des messages d'erreur. La première valeur, appelée *niveau journal de la console* (console loglevel), spécifie la plus basse priorité de messages qui sera affichée sur la console (veuillez noter que plus la priorité est basse, plus le numéro du niveau journal est élevé). La deuxième valeur définit le niveau journal par défaut pour les messages dépourvus de niveau journal explicite. La troisième valeur spécifie la plus basse configuration de niveau journal possible pour le niveau journal de la console. La dernière valeur définit la valeur par défaut pour le niveau journal de la console.
- rtsig-max — configure le nombre maximum de signaux POSIX en temps réel que le système peut avoir mis simultanément en file d'attente. La valeur par défaut est 1024.
- rtsig-nr — indique le nombre actuel de signaux POSIX en temps réel mis en file d'attente par le noyau.
- sem — représente le fichier configurant les paramètres de sémaphores dans le noyau. Un *sémaphore* est un objet IPC System V utilisé pour contrôler l'utilisation d'un processus spécifique.
- shmall — définit la quantité totale de mémoire partagée, en octets, qui peut être utilisée à un moment précis sur le système. Par défaut, cette valeur est de 2097152.
- shmmax — définit la plus grande taille autorisée par le noyau d'un segment de mémoire partagée, en octets. Par défaut, cette valeur est de 33554432. Le noyau prend cependant en charge des valeurs beaucoup plus élevées.
- shmmni — définit le nombre maximum de segments de mémoire partagée pour l'ensemble du système. Par défaut, cette valeur est de 4096
- sysrq — active la touche d'interrogation système, si cette valeur est différente de la valeur par défaut, qui est de 0. Reportez-vous à la Section 5.3.9 afin d'obtenir des informations détaillées sur la touche d'interrogation système.
- threads-max — définit le nombre maximum d'unités d'exécution devant être utilisées par le noyau, avec une valeur par défaut de 2048.

- version — affiche la date et l'heure de la dernière compilation du noyau. Le premier champ dans ce fichier, par exemple #3, fait référence au nombre de fois que le noyau a été construit à partir de la source.

Le répertoire random stocke un certain nombre de valeurs relatives à la génération de numéros aléatoires pour le noyau.

5.3.9.4. /proc/sys/net/

Ce répertoire contient des répertoires variés relatifs à des éléments réseau. Diverses configurations lors de la compilation du noyau déterminent la présence ou non de différents répertoires à cet endroit, comme par exemple appletalk, ethernet, ipv4, ipx et ipv6. Dans ces répertoires, vous pouvez ajuster les diverses valeurs réseau pour cette configuration sur un système en cours d'exécution.

Étant donné le nombre important d'options réseau possibles et disponibles sous Linux, ainsi que la grande quantité d'espace nécessaire pour en parler, nous n'aborderons que les répertoires /proc/sys/net/ les plus courants.

Le répertoire /proc/sys/net/core/ contient une série de paramètres qui contrôlent l'interaction entre le noyau et les couches réseau. Les fichiers les plus importants de ce répertoire sont:

- message_burst — la durée, en dixièmes de seconde, nécessaire pour écrire un nouveau message d'avertissement. Ceci est utilisé pour empêcher les attaques de *refus de service* (*DoS*). La valeur par défaut est de 50.
- message_cost — aussi utilisé pour empêcher les attaques de refus de service, en indiquant un coût sur chaque message d'avertissement. Plus la valeur de ce fichier est élevée (5 par défaut), plus il est probable que le message d'avertissement sera ignoré.

  L'idée de base d'une attaque DoS est de bombarder votre système de requêtes qui génèrent des erreurs et remplissent les partitions de disque de fichiers journaux ou qui accaparent toutes les ressources de votre système pour gérer la journalisation des erreurs. Les paramètres de message_burst et message_cost sont conçus pour être modifiés en fonction des risques acceptables de votre système par rapport au besoin d'une journalisation exhaustive.
- netdev_max_backlog — définit le nombre maximum de paquets pouvant être mis en file d'attente lorsqu'une interface spécifique reçoit des paquets plus rapidement que le noyau ne peut les traiter. La valeur par défaut de ce fichier est de 300.
- optmem_max — configure la taille maximum des tampons auxiliaires autorisée par socket.
- rmem_default — définit la taille par défaut en octets du tampon de socket de réception.
- rmem_max — définit la taille maximum en octets du tampon de réception.
- wmem_default — définit la taille par défaut en octets du tampon d'envoi.
- wmem_max — définit la taille maximum en octets du tampon d'envoi.

Le répertoire /proc/sys/net/ipv4/ contient des paramètres de mise en réseau supplémentaires. Bon nombre de ces paramètres, utilisés en connexion les uns avec les autres, sont très utiles pour empêcher des attaques contre votre système ou pour utiliser le système en tant que routeur.



Attention

Une modification inappropriée de ces fichiers pourrait avoir un effet néfaste sur votre connectivité distante au système.

Ci-dessous sont énumérés certains des fichiers les plus importants du répertoire /proc/sys/net/ipv4/:

- icmp_destunreach_rate, icmp_echoreply_rate, icmp_paramprob_rate et icmp_timeexeed_rate — définissent le délai maximum d'envoi, en centièmes de seconde, de paquets ICMP aux hôtes sous certaines conditions. La valeur 0 éliminant tout délai, elle n'est pas recommandée.
- icmp_echo_ignore_all et icmp_echo_ignore_broadcasts — permet au noyau d'ignorer les paquets ECHO ICMP de tous les hôtes ou uniquement ceux qui proviennent, respectivement, d'adresses de diffusion ou de multidiffusion. Une valeur de 0 permet au noyau de répondre, alors qu'une valeur de 1 elle, lui fait ignorer les paquets.
- ip_default_ttl — définit la *durée de vie (TTL)* (de l'anglais, 'Time To Live') par défaut, qui limite le nombre de sauts qu'un paquet peut faire avant d'atteindre sa destination. L'augmentation de cette valeur peut réduire les performances du système.
- ip_forward — permet aux interfaces du système de réacheminer des paquets aux autres interfaces. Par défaut, ce fichier est défini sur 0. En paramétrant ce fichier sur 1 vous activez le réacheminement des paquets réseau.
- ip_local_port_range — spécifie la plage de ports que TCP ou UDP doivent utiliser lorsqu'un port local est requis. Le premier nombre correspond au port le plus bas à utiliser et le second au port le plus élevé. Tout système sur lequel on s'attend à un nombre de ports requis supérieur aux valeurs 1024 à 4999 par défaut devrait utiliser la plage 32768 à 61000 dans ce fichier.
- tcp_syn_retries — fournit une limite du nombre de fois que votre système retransmet un paquet SYN lorsqu'il essaie d'effectuer une connexion.
- tcp_retries1 — définit le nombre de retransmissions permises, essayant de répondre à une connexion entrante. 3 est la valeur par défaut.
- tcp_retries2 — définit le nombre de retransmissions permises de paquets TCP. 15 est la valeur par défaut.

Le /usr/src/linux-2.4/Documentation/networking/ip-sysctl.txt contient une liste exhaustive des fichiers ainsi que des options disponibles dans le répertoire /proc/sys/net/ipv4/.

De nombreux autres répertoires existent dans le répertoire /proc/sys/net/ipv4/ et couvrent des sujets spécifiques. Le répertoire /proc/sys/net/ipv4/conf/ permet de configurer chaque interface du système de façon différente et d'utiliser des paramètres par défaut pour des périphériques non configurés (dans le sous-répertoire /proc/sys/net/ipv4/conf/default/) ainsi que des paramètres qui annulent toutes les configurations spéciales (dans le sous-répertoire /proc/sys/net/ipv4/conf/all/).

Le répertoire /proc/sys/net/ipv4/neigh/ contient non seulement des paramètres nécessaires pour la communication avec un hôte connecté directement au système (que l'on appelle voisin réseau) mais également des paramètres relatifs aux systèmes qui se trouvent à plusieurs sauts de distance.

Le routage via IPV4 dispose également de son propre répertoire, appelé /proc/sys/net/ipv4/route/. Contrairement à conf/ et neigh/, le répertoire /proc/sys/net/ipv4/route/ contient des spécifications qui s'appliquent au routage avec toutes les interfaces du système. Bon nombre de ces paramètres, tels que max_size, max_delay et min_delay, font référence au contrôle de la taille du cache de routage. Pour libérer le cache de routage, spécifiez simplement une valeur quelconque dans le fichier flush.

Des informations supplémentaires sur ces répertoires et les valeurs possibles pour leurs fichiers de configuration sont disponibles dans /usr/src/linux-2.4/Documentation/filesystems/proc.txt.

5.3.9.5. /proc/sys/vm/

Ce répertoire facilite la configuration du sous-système de la mémoire virtuelle (VM) du noyau Linux. Le noyau utilise de façon exhaustive et intelligente la mémoire virtuelle, que l'on appelle communément l'espace swap.

Les fichiers suivants se trouvent généralement dans le répertoire /proc/sys/vm/:

- bdflush — définit différentes valeurs liées au démon noyau bdflush.
- buffermem — permet de contrôler la quantité en pourcentage de la mémoire totale du système devant être utilisée comme mémoire tampon. La sortie de ce fichier ressemble à l'extrait ci-dessous:

  ```
  2       10      60
  ```

  La première et la dernière valeur définissent le pourcentage minimum et maximum de mémoire à utiliser comme mémoire tampon. La valeur au milieu indique le pourcentage de mémoire système dédié à la mémoire tampon à partir duquel le sous-système de gestion de la mémoire commencera à libérer davantage le cache tampon que les autres types de mémoire pour compenser le manque général de mémoire libre.
- kswapd — définit différentes valeurs relatives au démon de permutation du noyau kswapd. Ce fichier contient trois valeurs, à savoir:

  ```
  512 32 8
  ```

  La première valeur indique le nombre maximum de pages que kswapd essaiera de libérer en une seule tentative. Plus cette valeur est élevée, plus le noyau peut agir rapidement pour libérer des pages. La deuxième valeur définit le nombre minimum d'essais de libération d'une page par kswapd. La troisième valeur indique le nombre de pages que kswapd essaie d'écrire en une seule tentative. Un réglage précis de la valeur finale permet d'améliorer les performances des systèmes qui utilisent beaucoup d'espace swap en indiquant au noyau d'écrire les pages en blocs de grande taille, ce qui minimise le nombre de recherches disque.
- max_map_count — configure le nombre maximum de zones de topographie mémoire qu'un processus peut avoir. La valeur par défaut de 65536 est appropriée dans la plupart des cas.
- overcommit_memory — lorsque sa valeur par défaut est 0 le noyau estime la quantité de mémoire disponible et fait échouer les requêtes qui sont de toute évidence invalides. Malheureusement, étant donné que la mémoire est allouée à l'aide d'un algorithme heuristique plutôt que précis, cela peut parfois provoquer une surcharge du système.

  Si overcommit_memory a la valeur 1, le risque de surcharge du système est accru, mais cela permet également de développer les performances au niveau des tâches nécessitant beaucoup de mémoire, telles que celles effectuées par certains logiciels scientifiques.

  Les deux options suivantes ont été ajoutées pour les clients qui souhaitent prendre moins de risque au niveau d'un surengagement de la mémoire. Donner à overcommit_memory la valeur 2 échoue si une demande de mémoire est supérieure à la moitié de la mémoire vive, plus le swap. Lui donner une valeur de 3 échoue si la demande de mémoire est supérieure à ce que le swap seul peut garder.
- pagecache — contrôle la quantité de mémoire utilisée par le cache de page. Les valeurs de pagecache sont exprimées en pourcentage et fonctionnent de façon semblable à buffermem pour appliquer des valeurs minimales et maximales de mémoire cache de page disponible.
- page-cluster — définit le nombre de pages lues en une seule tentative. La valeur par défaut est 4 et se rapporte en fait à 16 pages; cette valeur est adéquate pour la plupart des systèmes.
- pagetable_cache — contrôle le nombre de tables de pages mises en cache par processeur. La première et la deuxième valeur font respectivement référence au nombre minimal et maximal de tables de pages à ne pas prendre en compte.

Le fichier /usr/src/linux-2.4/Documentation/sysctl/vm.txt contient des informations supplémentaires sur ces divers fichiers.

5.3.10. /proc/sysvipc/

Ce répertoire contient des informations sur les ressources IPC System V. Les fichiers de ce répertoire concernent les appels IPC System V de messages (`msg`), sémaphores (`sem`) et mémoire partagée (`shm`).

5.3.11. /proc/tty/

Ce répertoire contient des informations sur les *périphériques tty* disponibles et actuellement utilisés sur le système. Appelés à l'origine *périphériques téléimprimeurs* (ou télétypes), tout terminal basé sur les caractères est un périphérique tty.

Sous Linux, il existe trois types différents de périphérique tty. Les *périphériques série* sont utilisés avec les connexions série, par exemple par modem ou câble série. Les *terminaux virtuels* créent la connexion console commune, telle que les consoles virtuelles disponibles lorsque vous appuyez sur [Alt]-[<F-key>] sur la console système. Les *pseudo-terminaux* créent une communication à double sens utilisée par certaines applications de niveau supérieur, telles que XFree86. Le fichier `drivers` une liste des périphériques tty actuellement utilisés:

```
serial               /dev/cua        5  64-127 serial:callout
serial               /dev/ttyS       4  64-127 serial
pty_slave            /dev/pts      136   0-255 pty:slave
pty_master           /dev/ptm      128   0-255 pty:master
pty_slave            /dev/ttyp       3   0-255 pty:slave
pty_master           /dev/pty        2   0-255 pty:master
/dev/vc/0            /dev/vc/0       4       0 system:vtmaster
/dev/ptmx            /dev/ptmx       5       2 system
/dev/console         /dev/console    5       1 system:console
/dev/tty             /dev/tty        5       0 system:/dev/tty
unknown              /dev/vc/%d      4    1-63 console
```

Le fichier `/proc/tty/driver/serial` répertorie les statistiques d'utilisation et l'état de chaque ligne tty série.

Pour que les périphériques tty puissent être utilisés de façon semblable aux périphériques réseau, le noyau Linux applique une *procédure de transmission* sur les périphériques. Cela permet au pilote de placer un type spécifique d'en-tête sur chaque bloc de données transmis via un périphérique donné; ainsi, l'extrémité distante de la connexion voit ce bloc de données comme un tout unique dans un flux de blocs de données. SLIP et PPP sont des procédures de transmission courantes et sont communément utilisées pour connecter des systèmes via un lien série.

Les procédures de transmission enregistrées sont stockées dans le fichier `ldiscs` et des informations détaillées sont disponibles dans le répertoire `ldisc`.

5.4. Utilisation de la commande sysctl

La commande `/sbin/sysctl` est utilisée pour afficher, définir et automatiser les paramètres du noyau dans le répertoire `/proc/sys/`.

Pour avoir un aperçu rapide de tous les paramètres configurables du répertoire `/proc/sys/`, entrez la commande `/sbin/sysctl -a` en étant connecté en tant que super-utilisateur (root). Vous obtiendrez ainsi une longue liste exhaustive dont un court extrait figure ci-dessous:

```
net.ipv4.route.min_delay = 2
kernel.sysrq = 0
kernel.sem = 250     32000     32     128
```

Il s'agit des mêmes informations de base que celles que vous verriez si vous visualisiez chaque fichier individuellement. La seule différence réside dans l'emplacement du fichier. Le fichier /proc/sys/net/ipv4/route/min_delay est signifié par net.ipv4.route.min_delay où les barres obliques de répertoire sont remplacées par des points et la partie proc.sys est implicite.

Il est possible d'utiliser la commande sysctl au lieu de echo pour affecter des valeurs aux fichiers modifiables du répertoire /proc/sys/. Par exemple, au lieu d'utiliser la commande:

```
echo 1 > /proc/sys/kernel/sysrq
```

vous pouvez utiliser la commande sysctl:

```
sysctl -w kernel.sysrq="1"
kernel.sysrq = 1
```

Ce type de réglage rapide de valeurs individuelles dans /proc/sys/ est certes pratique en phase de tests, mais ne fonctionne pas aussi bien sur un système de production car tous les paramètres spéciaux de /proc/sys/ sont perdus lors du redémarrage du système. Pour préserver les paramètres que vous souhaitez affecter de façon permanente à votre noyau, ajoutez-les au fichier /etc/sysctl.conf.

Chaque fois que le système démarre, le programme init exécute le script /etc/rc.d/rc.sysinit. Ce dernier contient une commande pour exécuter sysctl à l'aide de /etc/sysctl.conf afin de déterminer les valeurs transmises au noyau. Toute valeur ajoutée à /etc/sysctl.conf prendra effet à chaque démarrage du système.

5.5. Ressources supplémentaires

Vous trouverez ci-dessous des sources d'informations supplémentaires sur le système de fichiers proc.

5.5.1. Documentation installée

L'essentiel de la documentation la plus pertinente sur /proc/ se trouve sur votre système.

- /usr/src/linux-2.4/Documentation/filesystems/proc.txt — contient des informations variées, mais limitées, sur tous les aspects du répertoire /proc/.
- /usr/src/linux-2.4/Documentation/sysrq.txt — offre un aperçu des options de la touche d'interrogation système.
- /usr/src/linux-2.4/Documentation/sysctl/ — est un répertoire contenant un certain nombre d'astuces en relation avec sysctl, y compris comment modifier des valeurs en rapport avec le noyau (kernel.txt), accès aux systèmes de fichiers (fs.txt) et utilisation de la mémoire virtuelle (vm.txt).
- /usr/src/linux-2.4/Documentation/networking/ip-sysctl.txt — Un examen de certaines option de mise en réseau d'IP.
- /usr/src/linux-2.4/ — vous trouverez les informations les plus pertinentes sur /proc/ en lisant le code source du noyau. Assurez-vous que le RPM kernel-source est installé sur votre système et consultez le répertoire /usr/src/linux-2.4/ pour le code source lui-même.

5.5.2. Sites Web utiles

- http://www.linuxhq.com — Ce site contient une base de données complète sur la source, les correctifs et la documentation de nombreuses versions du noyau Linux.





Chapitre 6. Utilisateurs et groupes

Le contrôle des *utilisateurs* et *groupes* est au coeur de l'administration de système de Red Hat Linux.

Les *utilisateurs* peuvent être aussi bien des personnes, avec des comptes attachés à des utilisateurs physiques, que des comptes existant pour une utilisation par des applications spécifiques

Les *groupes* sont des expressions logiques d'une société, regroupant des utilisateurs pour un but commun. Les utilisateurs appartenant à un groupe donné peuvent lire, écrire ou exécuter des fichiers appartenant à ce groupe.

Chaque utilisateur et chaque groupe se voit attribuer un numéro identificateur numérique unique appelé respectivement un *userid* (*UID*) et un *groupid* (*GID*).

Lors de sa création, tout fichier se voit assigner un utilisateur et un groupe. Il reçoit également les permissions de lecture, d'écriture et d'exécution pour le propriétaire du fichier, le groupe ou tout autre utilisateur. L'utilisateur et le groupe possédant un fichier, ainsi que les permissions d'accès à ce fichier, peuvent être modifiés par le super-utilisateur (ou root) et, dans la plupart des cas, par le créateur du fichier.

La bonne gestion des utilisateurs et groupes d'une part, et celle des permissions de fichiers d'autre part, font partie des tâches les plus importantes qu'un administrateur de système doive effectuer. Pour des informations plus détaillées sur les stratégies de gestion des utilisateurs et groupes, reportez-vous au chapitre intitulé *Managing Accounts and Group* (Gestion de comptes et groupe) du *Guide d'administration système de Red Hat Linux*.

6.1. Outils de gestion des utilisateurs et des groupes

La gestion des utilisateurs et des groupes peut être une tâche laborieuse, mais avec Red Hat Linux vous disposez des outils et conventions facilitant cette gestion.

La manière la plus simple de gérer des utilisateurs et groupes consiste à utiliser l'application graphique **Gestionnaire d'utilisateurs** (`redhat-config-users`). Pour plus d'informations sur le **Gestionnaire d'utilisateurs**, reportez-vous au chapitre intitulé *Configuration des utilisateurs et des groupes* du *Guide de personnalisation de Red Hat Linux*.

Les outils de la ligne de commande suivants peuvent également servir à gérer les utilisateurs et groupes:

- `useradd`, `usermod` et `userdel` — méthodes conformes aux standards de l'industrie permettant d'ajouter, de supprimer et modifier des comptes d'utilisateurs.
- `groupadd`, `groupmod` et `groupdel` — méthodes conformes aux standards de l'industrie permettant d'ajouter, de supprimer et modifier des groupes d'utilisateurs.
- `gpasswd` — méthode conforme aux standards de l'industrie permettant d'administrer le fichier `/etc/group`.
- `pwck`, `grpck` — outils permettant de vérifier le mot de passe, le groupe et les fichiers masqués associés.
- `pwconv`, `pwunconv` — outils permettant la conversion de mots de passe standard en mots de passe masqués et vice versa.

Pour un aperçu de la gestion d'utilisateurs et de groupes, reportez-vous au *Guide d'administration système de Red Hat Linux*. Pour des informations plus détaillées sur les outils de la ligne de commande permettant de gérer les utilisateurs et groupes, reportez-vous au chapitre intitulé *Configuration des utilisateurs et des groupes* du *Guide de personnalisation de Red Hat Linux*.

6.2. Utilisateurs standard

Dans le Tableau 6-1 sont énumérés les utilisateurs standard configurés dans le fichier /etc/passwd par une installation de type 'Complet' (Everything). L'identificateur groupe (ID groupe ou GID) figurant dans ce tableau correspond au *groupe primaire* pour l'utilisateur. Reportez-vous à la Section 6.3 pour une liste des groupes standard.

Utilisateur	UID	GID	Répertoire personnel	Shell
root	0	0	/root	/bin/bash
bin	1	1	/bin	/sbin/nologin
démon	2	2	/sbin	/sbin/nologin
adm	3	4	/var/adm	/sbin/nologin
lp	4	7	/var/spool/lpd	/sbin/nologin
sync	5	0	/sbin	/bin/sync
arrêt	6	0	/sbin	/sbin/shutdown
halt	7	0	/sbin	/sbin/halt
message	8	12	/var/spool/mail	/sbin/nologin
informations	9	13	/var/spool/news	
uucp	10	14	/var/spool/uucp	/sbin/nologin
opérateur	11	0	/root	/sbin/nologin
jeux	12	100	/usr/games	/sbin/nologin
gopher	13	30	/usr/lib/gopher-data	/sbin/nologin
ftp	14	50	/var/ftp	/sbin/nologin
personne	99	99	/	/sbin/nologin
rpm	37	37	/var/lib/rpm	/bin/bash
vcsa	69	69	/dev	/sbin/nologin
ntp	38	38	/etc/ntp	/sbin/nologin
canna	39	39	/var/lib/canna	/sbin/nologin
nscd	28	28	/	/bin/false
rpc	32	32	/	/sbin/nologin
postfix	89	89	/var/spool/postfix	/bin/true
nommé	25	25	/var/named	/bin/false
amanda	33	6	var/lib/amanda/	/bin/bash
postgres	26	26	/var/lib/pgsql	/bin/bash
sshd	74	74	/var/empty/sshd	/sbin/nologin
rpcuser	29	29	/var/lib/nfs	/sbin/nologin

Utilisateur	UID	GID	Répertoire personnel	Shell
nsfnobody	65534	65534	/var/lib/nfs	/sbin/nologin
pvm	24	24	/usr/share/pvm3	/bin/bash
apache	48	48	/var/www	/bin/false
xfs	43	43	/etc/X11/fs	/sbin/nologin
desktop	80	80	/var/lib/menu/kde	/sbin/nologin
gdm	42	42	/var/gdm	/sbin/nologin
mysql	27	27	/var/lib/mysql	/bin/bash
webalizer	67	67	/var/www/html/usage	/sbin/nologin
mailman	41	41	/var/mailman	/bin/false
mailnull	47	47	/var/spool/mqueue	/sbin/nologin
smmsp	51	51	/var/spool/mqueue	/sbin/nologin
squid	23	23	/var/spool/squid	/dev/null
ldap	55	55	/var/lib/ldap	/bin/false
netdump	34	34	/var/crash	/bin/bash
pcap	77	77	/var/arpwatch	/sbin/nologin
ident	98	98	/	/sbin/nologin
privoxy	100	101	/etc/privoxy	
radvd	75	75	/	/bin/false
fax	78	78	/var/spool/fax	/sbin/nologin
wnn	49	49	/var/lib/wnn	/bin/bash

Tableau 6-1. Utilisateurs standards

6.3. Groupes standard

Dans le Tableau 6-2 sont énumérés les groupes standards configurés par une installation de type 'Complet' (Everything). Sous Red Hat Linux, les groupes sont stockés dans le fichier /etc/group.

Groupe	GID	Membres
root	0	root
bin	1	root, bin, démon
démon	2	root, bin, démon
sys	3	root, bin, adm
adm	4	root, adm, démon
tty	5	
disque	6	root
lp	7	démon, lp

Groupe	GID	Membres
mem	8	
kmem	9	
wheel	10	root
message	12	message
informations	13	informations
uucp	14	uucp
man	15	
jeux	20	
gopher	30	
dip	40	
ftp	50	
verrouillage	54	
personne	99	
utilisateurs	100	
rpm	37	rpm
utmp	22	
disquette	19	
vcsa	69	
ntp	38	
canna	39	
nscd	28	
rpc	32	
postdrop	90	
postfix	89	
nommé	25	
postgres	26	
sshd	74	
rpcuser	29	
nfsnobody	65534	
pvm	24	
apache	48	
xfs	43	
desktop	80	
gdm	42	
mysql	27	

Groupe	GID	Membres
webalizer	67	
mailman	41	
mailnull	47	
smmsp	51	
squid	23	
ldap	55	
netdump	34	
pcap	77	
ident	98	
privoxy	101	
radvd	75	
fax	78	
slocate	21	
wnn	49	

Tableau 6-2. Groupes standards

6.4. Groupes propres à l'utilisateur

Red Hat Linux utilise un système de *groupe propre à l'utilisateur* (ou *UPG* de l'anglais 'User Private Group') qui facilite considérablement la gestion de groupes UNIX.

Un UPG est créé chaque fois qu'un nouvel utilisateur est ajouté au système. Les UPG portent le même nom que l'utilisateur pour lequel ils ont été créés et seul cet utilisateur est un membre de l'UPG.

Grâce à l'utilisation d'UPG, des permissions par défaut peuvent être déterminées en toute sécurité sur un nouveau fichier ou répertoire qui permettent à l'utilisateur, ainsi qu'au *groupe de cet utilisateur* de modifier le fichier ou répertoire.

Le paramètre qui détermine les différentes permissions à accorder à de nouveaux fichiers ou répertoires s'appelle *umask* et est configuré dans le fichier `/etc/bashrc`. Sur des systèmes UNIX, `umask` a traditionnellement une valeur de `022`, permettant uniquement l'utilisateur qui a créé le fichier ou répertoire de le modifier. Sous ce système, tous les autres utilisateurs, *y compris les membres du groupe du créateur*, ne sont pas autorisés à apporter de modifications. Cependant, étant donné que chaque utilisateur a son propre groupe privé dans le système UPG, cette "protection de groupe" n'est pas nécessaire.

6.4.1. Répertoire de groupes

De nombreuses sociétés du secteur informatique aiment créer un groupe pour chaque projet majeur et ensuite assignent des personnes aux groupes, si ces dernières doivent avoir accès aux fichiers du projet. Ce système traditionnel rend la gestion de fichiers difficile, car, lorsqu'une personne crée un fichier, ce dernier est associé au groupe primaire auquel la personne appartient. Lorsqu'une même personne travaille sur plusieurs projets, il devient difficile d'associer les bons fichiers au bon groupe. Toutefois, grâce système UPG, les groupes sont automatiquement assignés aux fichiers créés dans un répertoire avec le bit setgid défini, ce qui facilite considérablement la gestion des projets de groupe partageant un répertoire commun.

Supposons par exemple qu'un groupe de personnes travaille sur des fichiers figurant dans le répertoire /usr/lib/emacs/site-lisp/. Certaines personnes dignes de confiance peuvent certes être autorisées à modifier le répertoire, mais en aucun cas tout le monde. Ainsi, il faut d'abord créer un groupe emacs, comme le montre la commande suivante:

```
/usr/sbin/groupadd emacs
```

Afin d'associer le contenu du répertoire au groupe emacs, tapez:

```
chown -R root.emacs /usr/lib/emacs/site-lisp
```

Il est maintenant possible d'ajouter les utilisateurs appropriés au groupe à l'aide de la commande gpasswd:

```
/usr/bin/gpasswd -a <nom-d'utilisateur> emacs
```

Afin d'autoriser les utilisateurs à créer réellement des fichiers dans le répertoire, utilisez la commande suivante:

```
chmod 775 /usr/lib/emacs/site-lisp
```

Lorsqu'un utilisateur crée un nouveau fichier, il se voit assigner le groupe privé par défaut du groupe de l'utilisateur. Ensuite, donnez une valeur au bit setgid, qui donne à tout fichier créés dans le répertoire la même permission de groupe que le répertoire lui-même (emacs). Utilisez la commande suivante:

```
chmod 2775 /usr/lib/emacs/site-lisp
```

À ce stade, comme l'umask par défaut de chaque utilisateur est 002, tous les membres du groupe emacs peuvent créer et modifier des fichiers dans le répertoire /usr/lib/emacs/site-lisp/, sans que l'administrateur n'aie à changer les permissions de fichiers chaque fois que des utilisateurs enregistrent de nouveaux fichiers.

6.5. Mots de passe masqués

Dans un environnement multi-utilisateurs, il est primordial d'utiliser des *mots de passe masqués* (fournis par le paquetage shadow-utils et parfois appelés mots de passe ombre ou 'shadow passwords'). Ce faisant, la sécurité des fichiers d'authentification du système se voit accrue. Pour cette raison, le programme d'installation Red Hat Linux active des mots de passe masqués par défaut.

Ci-dessous figure une liste des avantages associés aux mots de passe masqués par rapport à l'ancienne manière de stocker des mots de passe sur des systèmes basés sur UNIX:

- amélioration de la sécurité du système en déplaçant les hachages de mots de passe cryptés d'un fichier /etc/passwd lisible par quiconque à un fichier /etc/shadow, seulement lisible par le super-utilisateur.
- stockage d'informations sur l'expiration de mots de passe.
- possibilité d'utiliser le fichier /etc/login.defs pour mettre en oeuvre les politiques de sécurité.

La plupart des utilitaires fournis par le paquetage shadow-utils fonctionnent correctement, que des mots de passe masqués soient activés ou non. Toutefois, comme les informations sur l'expiration des mots de passe sont stockées exclusivement dans le fichier /etc/shadow, aucune commande créant ou modifiant les informations sur l'expiration des mots de passe ne fonctionnera.

Ci-après figure une liste des commandes qui ne peuvent pas fonctionner sans que le mots de passe masqués ne soient préalablement activés:

- `chage`
- `gpasswd`
- `/usr/sbin/usermod` options `-e` ou `-f`
- `/usr/sbin/useradd` options `-e` ou `-f`





Chapitre 7. Le système X Window

Alors que le coeur de Red Hat Linux est son noyau, pour beaucoup d'utilisateurs, le visage du système d'exploitation est l'environnement graphique fourni par le *Système X Window*, aussi appelé tout simplement *X*.

De nombreux environnement de fenêtrage ont déjà existé dans le monde UNIX™ et ce, depuis des décennies, avant l'apparition de nombreux systèmes d'exploitations traditionnels courants. Au fil des années, X est devenu l'environnement graphique préféré des systèmes d'exploitation de type UNIX.

L'environnement graphique de Red Hat Linux est fourni par XFree86™, une implémentation Open Source de X. XFree86 est un projet de grande envergure, se développant rapidement grâce à des centaines de développeurs dans le monde entier. Il offre non seulement une prise en charge étendue pour un grand nombre de périphériques et d'architectures mais a également la capacité de tourner sur différents systèmes d'exploitation et plates-formes.

Le système X Window utilise une architecture client-serveur. Le *serveur X* reçoit les connexions d'applications *client X client* par le biais d'un réseau ou d'une interface de boucle locale. Le serveur communique avec le matériel, comme la carte vidéo, le moniteur, le clavier et la souris. Le client X se situe lui dans l'espace utilisateur, créant une *interface utilisateur graphique* (ou *GUI* de l'anglais 'Graphical User Interface') pour cet utilisateur et transmettant ses requêtes au le serveur .

7.1. XFree86

Red Hat Linux 9 utilise la version 4.x de XFree86 comme le système X Window de base, incluant de nombreux développements de pointe de la technologie XFree86, comme la prise en charge de l'accélération matérielle 3D, l'extension XRender pour de polices lissées, une conception basée sur des pilotes modulaire et une prise en charge du matériel vidéo moderne et des périphériques d'entrée.



Important

Red Hat Linux ne fournit plus de paquetages serveur XFree86 version 3. Avant d'effectuer une mise à niveau vers la dernière version de Red Hat Linux, assurez-vous que la carte vidéo est bien compatible avec la version 4 de XFree86 en consultant la liste de compatibilité du matériel de Red Hat disponible en ligne à l'adresse suivante: http://hardware.redhat.com.

Les fichiers concernant XFree86 se trouvent essentiellement dans deux emplacements:

/usr/X11R6/

contient le serveur X et certaines applications client ainsi que les fichiers d'en-tête, bibliothèques, modules et documentation de X.

`/etc/X11/`

contient tous les fichiers de configuration pour des applications client et serveur X. Ceci inclue les fichiers de configuration du serveur X lui-même, l'ancien serveur de polices `xfs`, les gestionnaires d'affichage X et bien d'autres composants de base.

Il est important de noter ici que le fichier de configuration pour la nouvelle architecture de polices basée sur Fontconfig est `/etc/fonts/fonts.conf` (qui remplace le fichier `/etc/X11/XftConfig`). Pour de plus amples informations sur la configuration et l'ajout de polices, reportez-vous à la Section 7.4.

Étant donné que le serveur X accomplit beaucoup de tâches difficiles en utilisant une large gamme de matériel, il nécessite une configuration détaillée. Le programme d'installation de Red Hat Linux met en place et configure XFree86 automatiquement, à moins que les paquetages XFree86 ne soient sélectionnés comme devant être installés. Toutefois, si le moniteur ou la carte vidéo changent, XFree86 devra être reconfiguré. Pour ce faire, la meilleure façon consiste à utiliser l'**Outil de configuration X** (`redhat-config-xfree86`).

Pour lancer l'**Outil de configuration X** pendant une session active de X, allez au bouton **Menu principal** (sur le panneau) => **Paramètres de système** => **Affichage**. Après une utilisation de l'**Outil de configuration X** pendant une session X, il faudra fermer la session X en cours, puis redémarrer X pour que les changements prennent effet. Pour obtenir de plus amples informations sur l'utilisation de L'**Outil de configuration X** reportez-vous au chapitre intitulé *Audio, Vidéo et divertissement en général* du *Guide de démarrage de Red Hat Linux*.

Dans certaines situations, il sera peut-être nécessaire de reconfigurer manuellement le serveur XFree86 en éditant son fichier de configuration `/etc/X11/XF86Config`. Pour obtenir de plus amples informations sur la structure de ce fichier, reportez-vous à la Section 7.3.

7.2. Environnements de bureau et gestionnaires de fenêtre

Une fois qu'un serveur XFree86 est en cours d'exécution, les application client X peuvent s'y connecter et créer une GUI pour l'utilisateur. Avec Red Hat Linux, une certaine variété de GUI est disponible, de l'interface rudimentaire du gestionnaire de fenêtre *Tab Window Manager* à celle hautement sophistiquée et interactive de l'environnement de bureau *GNOME*, auxquelles la plupart des utilisateurs de Red Hat Linux sont habitués.

Afin de créer cette dernière interface la plus perfectionnée, deux catégories principales de d'applications clients X doivent être connectées au serveur XFree86: un *environnement de bureau* et un *gestionnaire de fenêtre*.

7.2.1. Environnements de bureau

Un environnement de bureau rassemble des clients X assortis qui, lorsqu'ils sont utilisés ensemble, créent un environnement d'utilisateur graphique commun ainsi qu'une plateforme de développement.

Les environnements de bureau contiennent des fonctions plus avancées, qui permettent aux clients X et autres processus en cours, de communiquer les uns avec les autres. Ce faisant, toutes les applications écrites pour cet environnement peuvent s'intégrer parfaitement, comme par exemple la fonction de 'déplacement par glissement'.

Red Hat Linux fournit deux environnements de bureau:

- *GNOME* — L'environnement de bureau par défaut pour Red Hat Linux basé sur la boîte à outils graphique GTK+ 2.
- *KDE* — Un autre environnement de bureau basé sur la boîte à outils graphique Qt 3.

Aussi bien GNOME que KDE disposent non seulement d'applications de productivité avancées, comme des traitements de texte, des tableurs et des navigateurs Web mais fournissent également des outils permettant de personnaliser l'apparence de la GUI. De plus, si les deux bibliothèques GTK+ 2 et Qt sont installées, les applications de KDE peuvent être exécutées dans un environnement GNOME et vice versa.

Pour obtenir de plus amples informations sur la personnalisation des environnements de bureau GNOME et KDE, reportez-vous au *Guide de démarrage de Red Hat Linux*.

7.2.2. Gestionnaires de fenêtre

Les *gestionnaires de fenêtre* sont des programmes clients X qui font partie d'un environnement de bureau ou, dans certains cas, sont des applications à part entière. Leur objectif principal est de contrôler le positionnement, le redimensionnement et le déplacement des fenêtres graphiques. Les gestionnaires de fenêtre contrôlent également les barres de titres, le comportement du focus de la fenêtre et les liaisons personnalisées de touche et de souris.

Cinq gestionnaires de fenêtre sont compris dans Red Hat Linux:

- `kwin` — Le gestionnaire de fenêtre *KWin* est le choix par défaut pour l'environnement de bureau KDE. Il s'agit d'un gestionnaire simple et efficace qui supporte des thèmes personnalisés.
- `metacity` — Le gestionnaire de fenêtre *Metacity* est le choix par défaut pour l'environnement de bureau GNOME. Il s'agit d'un gestionnaire simple et efficace qui supporte des thèmes personnalisés.
- `mwm` — Le gestionnaire de fenêtre *Motif* est un gestionnaire de fenêtre à part entière dotés de fonctions élémentaires. Étant donné qu'il est supposé être un gestionnaire de fenêtre à part entière, il ne devrait pas être utilisé de concert avec les environnements de bureau GNOME ou KDE.
- `sawfish` — Le gestionnaire de fenêtre *Sawfish* est un gestionnaire de fenêtre dotés de nombreuses fonctions qui, jusqu'à la version 8.0 de Red Hat Linux était le choix par défaut pour l'environnement de bureau GNOME. Il peut être utilisé de concert avec un environnement de bureau ou en tant que gestionnaire de fenêtre à part entière.
- `twm` — Le gestionnaire de fenêtre minimaliste *Tab Window Manager*, fournissant l'outillage le plus élémentaire de tous les gestionnaires de fenêtre, peut être utilisé de concert avec un environnement de bureau ou en tant que gestionnaire de fenêtre à part entière. Il est installé en tant que composant de XFree86.

Ces gestionnaires de fenêtres peuvent fonctionner sans bureau afin de mieux se rendre compte de leurs différences. Pour ce faire, tapez la commande `xinit -e` <*chemin-du-gestionnaire-de-fenêtre*> où <*chemin-du-gestionnaire-de-fenêtre*> correspond à l'emplacement du fichier binaire du gestionnaire de fenêtre. Vous pourrez trouver ce fichier binaire en tapant `which` <*nom-du-gestionnaire-de-fenêtre*>.

7.3. Fichiers de configuration du serveur XFree86

Le serveur XFree86 un exécutable binaire (`/usr/X11R6/bin/XFree86`) qui charge dynamiquement au démarrage tous les modules de serveur X nécessaires depuis le répertoire `/usr/X11R6/lib/modules/`. Certains de ces modules sont automatiquement chargés par le serveur, alors que d'autres sont facultatifs et doivent donc être spécifiés dans le fichier de configuration du serveur XFree86.

Les fichiers de configuration du serveur XFree86 et les fichiers connexes sont stockés dans le répertoire `/etc/X11/`. Le fichiers de configuration du serveur XFree86 est `/etc/X11/XF86Config`. Quand Red Hat Linux est installé, les fichiers de configuration pour XFree86 sont créés en utilisant les informations relatives au matériel du système rassemblées lors du processus d'installation.

7.3.1. `XF86Config`

Bien qu'il soit rarement nécessaire de modifier manuellement le fichier de configuration `/etc/X11/XF86Config`, il est important de connaître les différentes sections et paramètres facultatifs qui existent. Ces connaissances sont particulièrement utiles lors de la résolution de problèmes.

7.3.1.1. La structure

Le fichier /etc/X11/XF86Config est composé d'un certain nombre de sections différentes qui traitent de certains aspects du matériel du système.

Chaque section commence par une ligne Section "*<nom-de-la-section>*" (où *<nom-de-la-section>* correspond au titre de la section) et finit par une ligne EndSection. Dans chacune des sections se trouvent des lignes contenant des noms d'option et au moins une valeur d'option, qui peut se trouver entre guillemets.

Les lignes commençant par un symbole dièse ([#]) ne sont pas lues par le serveur XFree86 et sont utilisées pour des commentaires en texte normal.

Certaines options contenues dans le fichier /etc/X11/XF86Config acceptent un commutateur booléen qui permet d'activer ou de désactiver l'option. Parmi les valeurs booléennes acceptables figurent:

- 1, on, true ou yes — Ces valeurs permettent d'activer l'option.
- 0, off, false ou no — Ces valeurs permettent de désactiver l'option.

La liste suivante explore certaines des sections les plus importantes, dans l'ordre dans lequel elles apparaissent dans un fichier /etc/X11/XF86Config typique. Des informations plus détaillées sur les fichiers de configuration du serveur XFree86 sont disponibles dans la page de manuel relative à XF86Config.

7.3.1.2. ServerFlags

La section facultative ServerFlags contient divers réglages globaux du serveur XFree86. Tout réglage dans cette section peuvent être remplacés par les options situées dans la section ServerLayout (reportez-vous à la Section 7.3.1.3 pour de plus amples informations).

Les entrées dans la section ServerFlags se trouvent sur leurs propres lignes et commencent par le terme Option et sont ensuite suivies d'une option spécifiée entre guillemets (["]).

Ci-dessous figure un exemple de section ServerFlags:

```
Section "ServerFlags"
      Option "DontZap" "true"
EndSection
```

Ci-dessous figure une liste de certaines des options les plus utiles:

- "DontZap" "*<booléen>*" — La valeur de *<booléen>* définie comme 'vrai' ('true') empêche l'utilisation de la combinaison de touches [Ctrl]-[Alt]-[Retour arrière] pour terminer instantanément le serveur XFree86.
- "DontZoom" "*<booléen>*" — La valeur de *<booléen>* définie comme 'vrai' ('true') empêche la commutation entre résolutions vidéos configurées par les combinaisons de [Ctrl]-[Alt]-[Plus] et [Ctrl]-[Alt]-[Moins].

7.3.1.3. ServerLayout

La section ServerLayout lie les périphériques d'entrée et de sortie contrôlés par le serveur XFree86. Au minimum, cette section doit spécifier un périphérique de sortie et au moins deux périphériques de sortie (un clavier et une souris).

L'exemple suivant illustre une section ServerLayout typique:

```
Section  "ServerLayout"
        Identifier     "Default Layout"
```

```
        Screen      0  "Screen0" 0 0
        InputDevice    "Mouse0" "CorePointer"
        InputDevice    "Keyboard0" "CoreKeyboard"
EndSection
```

Les entrées suivantes sont couramment utilisées dans la section `ServerLayout`:

- `Identifier` — spécifie un nom unique utilisé pour cette section `ServerLayout`.
- `Screen` — spécifie un nom d'une section `Screen` à utiliser avec le serveur XFree86. Il est possible de préciser plus d'une option `Screen`.

  Ci-dessous figure un exemple d'entrée `Screen` typique:

  ```
  Screen      0  "Screen0" 0 0
  ```

  Dans cette exemple d'entrée, le premier nombre `Screen` (`0`) indique que le premier connecteur du moniteur ou que la *tête* de la carte vidéo utilise la configuration spécifiée dans la section `Screen` avec l'identificateur `"Screen0"`.

  Si la carte vidéo a plus d'une tête, il faudra ajouter une entrée `Screen` avec un numéro différent et un identificateur différent pour la section `Screen`.

  Les nombres à la droite de `"Screen0"` donnent les coordonnées absolues X et Y pour le coin supérieur gauche de l'écran (par défaut `0 0`).
- `InputDevice` — spécifie le nom d'une section `InputDevice` à utiliser avec le serveur XFree86.

  Il doit y avoir au moins deux entrées `InputDevice`: une pour la souris par défaut et une pour le clavier par défaut. Les options `CorePointer` et `CoreKeyboard` indiquent qu'il s'agit du clavier et de la souris primaires.
- `Option "`*`<nom-de-l'option>`*`"` — correspond à une entrée facultative qui précise des paramètres supplémentaires pour cette section. Tout paramètre spécifié ici remplacent ceux mentionnés dans la section `ServerFlags`.

  Remplacez *`<nom-de-l'option>`* par une option valide pour cette section, parmi celles énumérées dans la page de manuel relative à XF86Config.

Il est possible de créer plus d'une section `ServerLayout`. Toutefois, le serveur ne lira que la section apparaissant en premier, à moins qu'une autre section `ServerLayout` ne soit spécifiée en tant qu'argument en ligne de commande.

7.3.1.4. `Files`

La section `Files` spécifie les chemins de services vitaux pour le serveur XFree86, comme le chemin des polices.

L'exemple suivant illustre une section `Files` typique:

```
Section "Files"
        RgbPath       "/usr/X11R6/lib/X11/rgb"
        FontPath      "unix/:7100"
EndSection
```

Parmi les entrées les plus communément utilisées dans la section `Files` figurent:

- `RgbPath` — spécifie l'emplacement de la base de données de couleurs RGB dans le système. Cette base de données définit tous les noms de couleurs valides dans XFree86 et les lie aux valeurs RGB particulières.
- `FontPath` — spécifie l'endroit où le serveur XFree86 doit se connecter pour obtenir les polices du serveur de polices `xfs`.

Par défaut, la valeur de `FontPath` est `unix/:7100`. Ceci indique au serveur XFree86 d'obtenir des informations de police en utilisant les connecteurs de domaine UNIX pour les communications inter-processus (IPC) sur le port 7100.

Consultez la Section 7.4 pour obtenir de plus amples informations sur XFree86 et les polices.

- `ModulePath` — Un paramètre facultatif permettant de spécifier d'autres répertoires pour le stockage de modules du serveur XFree86.

7.3.1.5. `Module`

La section `Module` spécifie les modules du répertoire `/usr/X11R6/lib/modules/` devant être chargés par le serveur XFree86. Ces modules fournissent au serveur XFree86 des fonctionnalités supplémentaires.

L'exemple suivant illustre une section `Module` typique:

```
Section "Module"
 Load  "dbe"
 Load  "extmod"
 Load  "fbdevhw"
 Load  "glx"
 Load  "record"
 Load  "freetype"
 Load  "type1"
 Load  "dri"
EndSection
```

7.3.1.6. `InputDevice`

Chaque section `InputDevice` configure un périphérique d'entrée pour le serveur XFree86. La plupart des systèmes possèdent en général au moins deux sections `InputDevice`: clavier et souris.

L'exemple suivant illustre une section `InputDevice` typique:

```
Section "InputDevice"
 Identifier  "Mouse0"
 Driver      "mouse"
 Option     "Protocol" "IMPS/2"
 Option     "Device" "/dev/input/mice"
 Option     "Emulate3Buttons" "no"
EndSection
```

Parmi les entrées les plus communément utilisées dans la section `InputDevice` figurent:

- `Identifier` — spécifie un nom unique pour cette section `InputDevice`. Cette entrée est nécessaire.
- `Driver` — spécifie le nom du pilote de périphérique devant être chargé par XFree86 pour le périphérique.
- `Option` — spécifie des options nécessaires concernant le périphérique.

  Pour une souris, ces options sont généralement:

  - `Protocol` — spécifie le protocole utilisé par la souris, comme par exemple `IMPS/2`.
  - `Device` — spécifie l'emplacement du périphérique physique.
  - `Emulate3Buttons` — spécifie si une souris à deux boutons doit se comporter comme une souris à trois boutons lorsque les deux boutons sont pressés simultanément.

Consultez la page de manuel relative à XF86Config pour obtenir une liste des options valides pour cette section.

Par défaut, la section `InputDevice` comporte des commentaires pour permettre aux utilisateurs de configurer des options supplémentaires.

7.3.1.7. section `Monitor`

La section `Monitor` permet de configurer le type de moniteur utilisé par le système. Alors qu'une section `Monitor` est le minimum requis, il tout à fait possible d'en avoir d'autres pour chaque type de moniteur utilisé par l'ordinateur.

La meilleure façon de configurer un moniteur consiste à configurer X lors du processus d'installation ou à utiliser l'**Outil de configuration X** . Pour obtenir de plus amples informations sur l'utilisation de l'**Outil de configuration X** , reportez-vous au chapitre intitulé *Audio, Vidéo et divertissement en général* du *Guide de démarrage de Red Hat Linux*.

L'exemple suivant illustre une section `Monitor` typique pour un moniteur:

```
Section "Monitor"
 Identifier   "Monitor0"
 VendorName   "Monitor Vendor"
 ModelName    "DDC Probed Monitor - ViewSonic G773-2"
 DisplaySize  320 240
 HorizSync    30.0 - 70.0
 VertRefresh  50.0 - 180.0
EndSection
```

**Avertissement**

Faites très attention lorsque vous éditez manuellement les valeurs de la section `Monitor` de `/etc/X11/XF86Config`. En effet, l'utilisation de valeurs inappropriées dans cette section peuvent endommager ou même détruire votre moniteur. Consultez la documentation accompagnant votre moniteur pour connaître les paramètres acceptables disponibles.

Parmi les entrées les plus communément utilisées dans la section `Monitor` figurent:

- `Identifier` — spécifie un nom unique utilisé pour cette section `Monitor`. Cette entrée est nécessaire.
- `VendorName` — correspond à un paramètre facultatif précisant le nom du fabricant du moniteur.
- `ModelName` — correspond à un paramètre facultatif précisant le nom de modèle du moniteur.
- `DisplaySize` — correspond à un paramètre facultatif précisant en millimètres, la taille physique de la partie image du moniteur.
- `HorizSync` — spécifie la gamme de fréquences sync horizontales compatible avec le moniteur en kHz. Ces valeurs aident le serveur XFree86 à déterminer la validité des entrées `Modeline` prédéfinies ou spécifiées pour le moniteur.
- `VertRefresh` — spécifie la gamme des fréquences de rafraîchissement vertical prise en charge par le moniteur, en kHz. Ces valeurs aident le serveur XFree86 à déterminer la validité des entrées `Modeline` pré-définies ou spécifiées pour le moniteur.

- `Modeline` — correspond à un paramètre facultatif précisant les modes vidéo supplémentaires utilisés par le moniteur pour des résolutions particulières, avec certaines résolutions de rafraîchissement horizontal sync et vertical. Pour obtenir de plus amples explications sur les entrées `Modeline`, consultez la page de manuel relative à `XF86Config`.
- `Option "<nom-de-l'option>"` — correspond à une entrée facultative qui précise des paramètres supplémentaires pour cette section. Remplacez *<nom-de-l'option>* par une option valide pour cette section, selon celles énumérées dans la page de manuel relative à XF86Config.

7.3.1.8. `Device`

Chaque section `Device` configure section une carte vidéo utilisée par le système. Alors qu'une section `Device` est le minimum requis, il tout à fait possible d'en avoir d'autres pour chaque carte vidéo installée sur l'ordinateur.

La meilleure façon de configurer une carte vidéo consiste à configurer X lors du processus d'installation ou à utiliser l'**Outil de configuration X** . Pour obtenir de plus amples informations sur l'utilisation de l'**Outil de configuration X** , reportez-vous au chapitre intitulé *Audio, Video et divertissement en général* du *Guide de démarrage de Red Hat Linux*.

L'exemple suivant illustre une section `Device` typique pour une souris:

```
Section "Device"
 Identifier  "Videocard0"
 Driver      "mga"
 VendorName  "Videocard vendor"
 BoardName   "Matrox Millennium G200"
 VideoRam    8192
        Option      "dpms"
EndSection
```

Parmi les options les plus communément utilisées dans la section `Device` figurent:

- `Identifier` — spécifie un nom unique utilisé pour cette section `Device`. Cette entrée est nécessaire.
- `Driver` — spécifie le pilote devant être chargé par le serveur XFree86 afin que la carte vidéo puisse être utilisée. Une liste de pilotes est disponible dans le fichier `/usr/X11R6/lib/X11/Cards`, qui est installé avec le paquetage `hwdata`.
- `VendorName` — correspond à un paramètre facultatif précisant le nom du fabricant du moniteur.
- `BoardName` — correspond à un paramètre facultatif précisant le nom de la carte vidéo.
- `VideoRam` — correspond à un paramètre facultatif précisant quantité de mémoire RAM disponible sur la carte vidéo, en kilobits. Ce paramètre n'est nécessaire que pour les cartes vidéos que XFree86 ne peut pas détecter et pour lesquelles il ne peut donc pas correctement déterminer la quantité de RAM vidéo.
- `BusID` — correspond à une entrée facultative précisant l'emplacement du bus de la carte vidéo. Cette option n'est nécessaire que pour les systèmes dotés de cartes multiples.
- `Screen` — correspond à une entrée facultative précisant le connecteur du moniteur ou la tête de la carte vidéo que la section `Device` configure. Cette option n'est nécessaire que pour les cartes vidéo à têtes multiples.

  Si de multiples moniteurs sont connectés à des têtes différentes sur la même carte vidéo, il est nécessaire non seulement d'avoir des sections `Device` séparées mais chacune de ces sections doit également avoir une valeur `Screen` différente.

Les valeurs associées à l'entrée `Screen` doivent être entières. La première tête de la carte vidéo à une valeur de `0`. La valeur de chaque tête supplémentaire augmente d'une unité.

- `Option "<nom-de-l'option>"` — correspond à une entrée facultative qui précise des paramètres supplémentaires pour cette section. Remplacez *<nom-de-l'option>* par une des options valides pour cette section, énumérées dans la page de manuel relative à XF86Config.

  `"dpms"` est une des options très courantes permettant d'activer la conformité en alimentation à Service Star pour le moniteur.

7.3.1.9. `Screen`

Chaque section `Screen` lie une carte vidéo (ou tête de carte vidéo) à un moniteur en référençant la section `Device` et la section `Monitor` pour chaque. Bien qu'une section `Screen` soit le minimum requis, il est possible d'avoir d'autres instances pour chaque combinaison vidéo et moniteur existant sur l'ordinateur.

L'exemple suivant illustre une section `Screen` typique:

```
Section "Screen"
 Identifier "Screen0"
 Device     "Videocard0"
 Monitor    "Monitor0"
 DefaultDepth     16
 SubSection "Display"
  Depth     24
  Modes    "1280x1024" "1280x960" "1152x864" "1024x768" "800x600" "640x480"
 EndSubSection
 SubSection "Display"
  Depth     16
  Modes    "1152x864" "1024x768" "800x600" "640x480"
 EndSubSection
EndSection
```

Parmi les entrées les plus communément utilisées dans la section `Screen` figurent:

- `Identifier` — spécifie un nom unique utilisé pour cette section `Screen`. Cette entrée est nécessaire.
- `Device` — spécifie le nom unique d'une section `Device`. Cette entrée est nécessaire.
- `Monitor` — spécifie le nom unique d'une section `Monitor`. Cette entrée est nécessaire.
- `DefaultDepth` — spécifie l'intensité des couleurs par défaut, en bits. Dans l'exemple précédent, la valeur par défaut de `16` fournit des milliers de couleurs. Au moins une entrées `DefaultDepth` est nécessaire, mais de multiples entrées sont acceptées.
- `SubSection "Display"` — spécifie les modes écran disponibles à une intensité de couleur donnée. Bine qu'une section `Screen` puisse contenir de multiples sous-sections `Display`, il doit y en avoir au moins une pour l'intensité de couleur spécifiée dans l'entrée `DefaultDepth`.
- `Option "<nom-de-l'option>"` — correspond à une entrée facultative qui précise des paramètres supplémentaires pour cette section. Remplacez *<nom-de-l'option>* par une des options valides pour cette section, énumérées dans la page de manuel relative à XF86Config.

7.3.1.10. DRI

La section facultative `DRI` spécifie les paramètres pour *Direct Rendering Infrastructure* (*DRI*). DRI est une interface dont la fonction principale est de permettre aux applications logicielles 3D de profiter des capacités d'accélération 3D intégrés dans la plupart de matériel vidéos moderne. De plus, DRI peut améliorer les performances 2D grâce à l'accélération matérielle, dans le cas où elle serait prise en charge par le pilote de la carte vidéo.

Cette section n'est pas prise en compte à moins que l'interface DRI ne soit activée dans la section `Module`.

L'exemple suivant illustre une section `DRI` typique:

```
Section "DRI"
        Group        0
        Mode         0666
EndSection
```

Étant donné que différentes cartes vidéo utilise la DRI de manière différente, nous vous recommandons de ne pas changer les valeurs de cette section sans consulter d'abord le fichier `/usr/X11R6/lib/X11/doc/README.DRI`.

7.4. Polices

Red Hat Linux utilise deux méthodes pour gérer les polices et afficher sous XFree86. Le tout nouveau sous-système de polices Fontconfig simplifie le gestion des polices et fournit des fonctions d'affichage avancées, comme le lissage. Ce système est utilisé automatiquement pour des applications programmées pour utiliser la boîte à outils graphiques Qt 3 ou GTK+ 2.

Pour des raisons de compatibilité, Red Hat Linux fournit le sous-système de polices original appelé le sous-système de polices X de base ('core'). Ce système, qui a plus de 15 ans, s'articule autour du *Serveur de polices X* (*xfs*).

Cette section examine la configuration des polices pour X utilisant les deux systèmes.

7.4.1. Fontconfig

Le sous-système de polices Fontconfig permet à des applications d'accéder directement aux polices du système et utilise Xft ou tout autre mécanisme de rendu des polices de Fontconfig avec un lissage avancé. Des applications graphiques peuvent utiliser la bibliothèque Xft avec Fontconfig afin de créer du texte à l'écran.

Au fil du temps, les sous-système Fontconfig/Xft remplacera le sous-système de polices X de base.

**Important**

> Le sous-système de polices Fontconfig ne peut pas encore être utilisé avec **OpenOffice.org** et **Abiword**, qui utilisent leur propre technologie de rendu des polices.

Il est important de noter ici que Fontconfig partage le fichier de configuration `/etc/fonts/fonts.conf`, qui remplace le fichier `/etc/X11/XftConfig`. Le fichier de configuration de Fontconfig ne doit pas être modifié manuellement.



Astuce

En raison de la transition vers le nouveau système de polices, les applications GTK+ 1.2 ne sont affectées par aucun changement apportés par le bais du dialogue **Préférences de polices** (accessible en sélectionnant le bouton **Menu principal** [sur le panneau] => **Préférences** => **Polices**). Pour ces applications, une police peut être configurée en ajoutant les lignes suivantes au fichier `~/.gtkrc.mine`:

```
style "user-font" {

fontset = "<spécification-de-police>"

}

widget_class "*" style "user-font"
```

Remplacez *<spécification-de-police>* par la spécification de la police dans le style utilisé par les applications X classiques, comme par exemple, `-adobe-helvetica-medium-r-normal--*-120-*-*-*-*-*-*`. Il est possible d'obtenir une liste complète des polices de base en exécutant `xlsfonts` ou d'en créer une de manière interactive en utilisant `xfontsel`.

7.4.1.1. Ajout de polices à Fontconfig

L'ajout de nouvelles polices au sous-système Fontconfig est un processus relativement simple.

1. Pour ajouter de nouvelles polices pour tout le système, copiez les nouvelles polices dans le répertoire `/usr/share/fonts/local/`.

   Pour ajouter de nouvelles polices pour un utilisateur spécifique, copiez les nouvelles polices dans le répertoire `.fonts/` du répertoire personnel (ou home) de l'utilisateur.

2. Pour mettre à jour le cache des informations de polices, utilisez la commande `fc-cache` comme dans l'exemple suivant:

   `4fc-cache <chemin-vers-le-répertoire-de-polices>`

   Dans cette commande, remplacez *<chemin-vers-le-répertoire-de-polices>* par le répertoire contenant ces nouvelles polices (soit `/usr/share/fonts/local/` soit `~/.fonts/`).



Astuce

Des utilisateurs peuvent aussi installer des polices graphiquement de manière individuelle, en parcourant `fonts:///` dans Nautilus et en y faisant glisser les nouveaux fichiers de polices.



Important

Si le nom du fichier de polices finit par une extension `.gz`, il s'agit d'un fichier compressé qui ne pourra être utilisé à moins d'être décompressé. Pour ce faire, utilisez la commande `gunzip` ou cliquez deux fois sur le fichier et faites glisser la police vers un répertoire dans **Nautilus**.

7.4.2. Système de polices X de base

Pour des raisons de compatibilité, Red Hat Linux inclut toujours le sous-système de polices original appelé le sous-système de polices X de base ('core'), utilisant le serveur de polices X (`xfs`) pour fournir les polices aux applications clients X.

Le serveur XFree86 recherche un serveur de police spécifié dans l'entrée `FontPath` dans la section `Files` du fichier de configuration `/etc/X11/XF86Config`. Pour obtenir de plus amples informations sur l'entrée `FontPath`, reportez-vous à la Section 7.3.1.4.

Le serveur XFree86 se connecte au serveur `xfs` sur un port définit afin d'obtenir des informations sur les polices. Dans de telles circonstances, le service `xfs` doit être en cours d'exécution pour que X puisse démarrer. Pour obtenir de plus amples informations sur la configuration de services à un niveau d'exécution particulier, reportez-vous au chapitre intitulé *Contrôle de l'accès aux services* du *Guide de personnalisation de Red Hat Linux*.

7.4.2.1. Configuration de `xfs`

Le script `/etc/rc.d/init.d/xfs` lance le serveur `xfs`. Il est possible de configurer plusieurs options dans le fichier `/etc/X11/fs/config`.

Ci-dessous figure une liste des options les plus courantes:

- `alternate-servers` — spécifie une liste d'autres serveurs de polices à utiliser si ce serveur de polices n'est pas disponible. Chaque serveur dans cette liste doit être séparé par une virgule.
- `catalogue` — spécifie un liste ordonnée de chemins de police à utiliser. Chaque chemin de police doit être séparé par une virgule pour que la liste soit exploitable.

  Utilisez la chaîne `:unscaled` immédiatement après le chemin de polices pour faire charger en premier les polices non-proportionnées dans cette liste. Spécifiez alors à nouveau le chemin de police entier, pour que les autres polices proportionnées soient également chargées.
- `client-limit` — spécifie le nombre maximum de clients que ce serveur de polices va approvisionner. La valeur par défaut est `10`.
- `clone-self` — autorise le serveur de polices à reproduire une autre version de lui-même lorsque la limite de clients (`client-limit`) est atteinte. La valeur par défaut pour cette option est `on`.
- `default-point-size` — spécifie la taille de point par défaut pour toute police qui ne spécifie pas cette valeur. La valeur par défaut est en décipoints. La valeur par défaut de `120` correspond à une police de 12 points.
- `default-resolutions` — spécifie une liste de résolutions prises en charge par le serveur XFree86. Chaque résolution figurant dans la liste doit être séparée par une virgule.
- `deferglyphs` — spécifie si le chargement de *glyphs* (le graphique utilisé pour la représentation visuelle d'une police) doit être différé. Pour désactiver cette fonction, utilisez `none`; pour l'activer pour toutes ces polices, utilisez `all` ou pour ne l'activer que pour les polices 16-bit, utilisez `16`.
- `error-file` — spécifie le chemin et le nom du fichier de l'endroit où les erreurs `xfs` doivent être enregistrées.
- `no-listen` — empêche `xfs` d'être attentif à des protocoles spécifiques. Cette option à par défaut la valeur `tcp` afin d'empêcher `xfs` de recevoir des connexions sur les ports TCP, surtout pour des raisons de sécurité. Si vous utilisez `xfs` pour servir des polices à travers le réseau, supprimez cette ligne.
- `port` — spécifie le port TCP sur lequel `xfs` recevra des connexions si `no-listen` n'existe pas ou est désactivé par un commentaire.
- `use-syslog` — spécifie si le journal d'erreurs système doit être utilisé.

7.4.2.2. Ajout de polices à **xfs**

Pour ajouter des polices au sous-système de polices X de base (`xfs`), suivez les étapes suivantes:

1. À moins qu'il n'existe déjà, créez un répertoire nommé `/usr/share/fonts/local/` à l'aide de la commande suivante, en étant connecté en tant que super-utilisateur (ou root):
   ```
   mkdir /usr/share/fonts/local/
   ```
   Si la création du répertoire `/usr/share/fonts/local/` est nécessaire, il faut ajouter ce dernier au chemin `xfs` à l'aide de la commande suivante, en étant connecté en tant que super-utilisateur (ou root):
   ```
   chkfontpath --add /usr/share/fonts/local/
   ```
2. Copiez le nouveau fichier de polices dans le répertoire `/usr/share/fonts/local/`.
3. Mettez à jour les informations de polices à l'aide de la commande suivante, en étant connecté en tant que super-utilisateur (ou root):
   ```
   ttmkfdir -d /usr/share/fonts/local/ -o /usr/share/fonts/local/fonts.scale
   ```
4. Redémarrez le serveur de polices `xfs` à l'aide de la commande suivante, en étant connecté en tant que super-utilisateur (ou root):
   ```
   service xfs reload
   ```

7.5. Niveaux d'exécution et XFree86

Dans la plupart des cas, l'installation par défaut de Red Hat Linux configure l'ordinateur pour qu'il démarre dans un environnement de connexion graphique, connu en tant que niveau d'exécution 5. Il est toutefois possible de démarrer en mode multi-utilisateurs texte-seul, connu en tant que niveau d'exécution 3, et de démarrer ainsi un session X.

Pour obtenir de plus amples informations sur les niveaux d'exécution, reportez-vous à la Section 1.4.

Cette section passe en revue le démarrage de XFree86 aussi bien au niveau d'exécution 3 qu'au niveau d'exécution 5.

7.5.1. Niveau d'exécution 3

Au niveau d'exécution 3, la meilleure façon de lancer une session X consiste se connecter et à taper la commande `startx`. Cette commande `startx` est une commande frontale (ou 'front-end') à la commande `xinit` qui lance le serveur XFree86 et y connecte les applications client X. Étant donné que l'utilisateur est déjà connecté au système au niveau d'exécution 3, `startx` ne lance pas un gestionnaire d'affichage et n'authentifie pas les utilisateurs. Pour obtenir de plus amples informations sur les gestionnaires d'affichage, reportez-vous à la Section 7.5.2.

Lorsque la commande `startx` est exécutée, elle recherche un fichier `.xinitrc` dans le répertoire personnel (ou home) de l'utilisateur pour définir l'environnement de bureau et, le cas échéant, d'autres application client X à lancer. Si aucun fichier `.xinitrc` n'existe, il enclenchera à sa place le fichier `/etc/X11/xinit/xinitrc` par défaut du système.

Le script `xinitrc` par défaut recherche alors les fichiers définis par l'utilisateur et les fichiers systèmes par défaut, y compris `.Xresources`, `.Xmodmap` et `.Xkbmap` dans le répertoire personnel de l'utilisateur d'une part, et `Xresources`, `Xmodmap` et `Xkbmap` dans le répertoire `/etc/X11/` d'autre part. Les fichiers `Xmodmap` et `Xkbmap`, s'ils existent, sont utilisés par l'utilitaire `xmodmap` pour configurer le clavier. Les fichiers `Xresources` sont lus afin d'assigner des valeurs préférentielles spécifiques aux applications.

Après avoir paramétré ces options, le script xinitrc exécute tous les scripts situés dans le répertoire /etc/X11/xinit/xinitrc.d/. Parmi les scripts importants faisant partie de ce répertoire figure xinput, permettant de configurer des paramètres comme la langue par défaut.

Ensuite, le script xinitrc essaie d'exécuter .Xclients dans le répertoire personnel (home) de l'utilisateur et recoure à /etc/X11/xinit/Xclients s'il ne peut pas le trouver. Le rôle du fichier Xclients est de démarrer l'environnement de bureau ou, le cas échéant, un simple gestionnaire de fenêtre élémentaire. Le script .Xclients dans le répertoire personnel de l'utilisateur lance l'environnement de bureau spécifié par l'utilisateur dans le fichier .Xclients-default. Si le fichier .Xclients n'existe pas dans le répertoire personnel de l'utilisateur, le script standard /etc/X11/init/Xclients tente de lancer un autre environnement de bureau, en premier GNOME et en second KDE suivi de twm.

L'utilisateur revient à une session utilisateur en mode texte après s'être déconnecté de X au niveau d'exécution 3.

7.5.2. Niveau d'exécution 5

Lorsque le système démarre au niveau d'exécution 5, une application client X spéciale appelée gestionnaire d'affichage, est lancée. Un utilisateur doit s'authentifier en utilisant le gestionnaire d'affichage avant que tout environnement de bureau ou gestionnaire de fenêtre ne puisse être lancé.

Selon les environnements de bureaux installés sur le système, trois gestionnaires d'affichage différents sont disponibles pour assurer l'authentification de l'utilisateur.

- gdm — Le gestionnaire d'affichage par défaut pour Red Hat Linux; gdm permet la configuration des paramètres de langage, le démarrage, l'arrêt et la connexion au système par l'utilisateur.
- kdm — Le gestionnaire d'affichage de KDE permettant le démarrage, l'arrêt et la connexion au système par l'utilisateur.
- xdm — Un gestionnaire d'affichage rudimentaire ne permettant que la connexion de l'utilisateur au système.

Lors du démarrage au niveau d'exécution 5, le script prefdm détermine le gestionnaire d'affichage de préférence en consultant le fichier /etc/sysconfig/desktop. Pour obtenir une liste des options disponibles pour ce fichier, reportez-vous au fichier /usr/share/doc/initscripts-<*numéro-de-version*>/sysconfig.txt (où <*numéro-de-version*> correspond au numéro de version du paquetage initscripts).

Chacun des gestionnaires d'affichage consultent le fichier /etc/X11/xdm/Xsetup_0 pour configurer l'écran de connexion. Une fois que l'utilisateur s'est connecté au système, le script /etc/X11/xdm/GiveConsole s'exécuter pour assigner à l'utilisateur la propriété de la console. Ensuite, le script /etc/X11/xdm/Xsession se lance pour effectuer de nombreuses tâches habituellement exécutées par le script xinitrc lorsque X est démarré au niveau d'exécution 3, y compris le paramétrage du système et des ressources utilisateurs, et le lancement des scripts dans le répertoire /etc/X11/xinit/xinitrc.d/.

L'utilisateur peut spécifier l'environnement de bureau qu'il souhaite utiliser quand il s'authentifie par le bais des gestionnaires d'affichage gdm ou kdm en le sélectionnant dans le menu **Session** (accessible en choisissant le bouton **Menu principal** [sur le panneau] => **Préférences** => **Préférences supplémentaires** => **Sessions**). Si l'environnement de bureau n'est pas spécifié dans le gestionnaire de fenêtre, le script /etc/X11/xdm/Xsession vérifiera les fichiers .xsession et .Xclients dans le répertoire personnel (ou home) de l'utilisateur pour décider quel environnement de bureau charger. En dernier ressort, le fichier /etc/X11/xinit/Xclients la référence pour sélectionner un environnement de bureau ou gestionnaire de fenêtres à utiliser, de la même façon que pour le niveau d'exécution 3.

Lorsque l'utilisateur termine une session X sur l'affichage par défaut (:0) et se déconnecte, le script /etc/X11/xdm/TakeConsole s'exécute et réassigne la propriété de la console au super-utilisateur

(ou root). Le gestionnaire d'affichage original, qui ne s'est pas éteint depuis la connexion de l'utilisateur, reprend le contrôle déclenchant un nouveau gestionnaire d'affichage. Ce faisant, le serveur XFree86 est redémarré, un nouvel écran d'authentification est affiché et tout le processus recommence.

L'utilisateur revient au gestionnaire d'affichage après s'être déconnecté de X au niveau d'exécution 5.

Pour obtenir de plus amples informations sur le contôle de l'authentification des utilisateurs par les gestionnaires d'affichage, reportez-vous d'une part au fichier /usr/share/doc/gdm-<*numéro-de-version*>/README (où <*numéro-de-version*> correspond au numéro de version du paquetage gdminstallé) et d'autre part à la page de manuel relative à xdm.

7.6. Ressources supplémentaires

Il existe de nombreuses informations détaillées sur le serveur XFree86, les clients qui s'y connectent et sur les environnements de bureau et gestionnaires de fenêtre.

7.6.1. Documentation installée

- /usr/X11R6/lib/X11/doc/README — Décrit brièvement l'architecture XFree86 et la façon d'obtenir des informations supplémentaires sur le projet XFree86 en tant que nouvel utilisateur.
- /usr/X11R6/lib/X11/doc/RELNOTES — Pour les utilisateurs avancés qui veulent connaître les dernières fonctions offertes par XFree86.
- man XF86Config — Contient des informations sur les fichiers de configuration de XFree86, comprenant la signification et la syntaxe des différentes sections des fichiers.
- man XFree86 — La page de manuel principale sur XFree86, détaille les différences entre les connexions de serveur X locales et de réseau, explore les variables d'environnement courantes, énumère les options de ligne de commande et fournit des combinaisons de touches utiles pour l'administration.
- man Xserver — Décrit le serveur d'affichage X.

7.6.2. Sites Web utiles

- http://www.xfree86.org — Page d'accueil du projet XFree86, qui produit la version XFree86 Open Source du système X Window. XFree86 est livré avec Red Hat Linux pour contrôler le matériel nécessaire et fournir un environnement d'interface graphique (GUI).
- http://dri.sourceforge.net — Page d'accueil du projet DRI (Direct Rendering Infrastructure). La DRI est le composant central de l'accélération du matériel 3D pour XFree86.
- http://www.redhat.com/mirrors/LDP/HOWTO/XFree86-HOWTO — Un document HOWTO détaillant le manuel d'installation et la configuration personnalisée de XFree86.
- http://www.gnome.org/ — Page d'accueil du projet GNOME.
- http://www.kde.org/ — Page d'accueil de l'environnement de bureau KDE.
- http://nexp.cs.pdx.edu/fontconfig/ — Page d'accueil du sous-système de polices Fontconfig pour XFree86.

7.6.3. Livres sur le sujet

- *The Concise Guide to XFree86 for Linux* de Aron Hsiao; Que — Fournit l'avis d'un expert sur le fonctionnement de XFree86 sur les systèmes Linux.
- *The New XFree86* de Bill Ball; Prima Publishing — Examine XFree86 et sa relation avec les environnements de bureau couramment utilisés, comme GNOME et KDE.
- *Beginning GTK+ and GNOME* de Peter Wright; Wrox Press, Inc. — Présente aux programmeurs l'architecture GNOME, leur montrant comment débuter dans GTK+.
- *GTK+/GNOME Application Development* de Havoc Pennington; New Riders Publishing — Un examen avancé au coeur de la programmation GTK+, concentré sur un échantillon de code et une étude exhaustive des API disponibles.
- *KDE 2.0 Development* de David Sweet et Matthias Ettrich; Sams Publishing — Expose aux développeurs débutants et avancés comment exploiter au maximum les nombreuses directives d'environnement nécessaires à l'élaboration d'applications QT pour KDE.

II. Références aux services du réseau

Sous Red Hat Linux, il est possible de déployer un grand éventail de services réseau. Cette partie décrit non seulement la manière dont les interfaces réseau sont configurées mais fournit également des informations détaillées sur des services réseau critiques tels que NFS, Serveur HTTP Apache, Sendmail, Fetchmail, Procmail, BIND et LDAP.

Table des matières



Chapitre 8.
Interfaces réseau

Sous Red Hat Linux, toutes les communications réseau se font entre des *interfaces* logicielles configurées et des périphériques réseau connectés au système.

Les fichiers de configuration pour les interfaces réseau et les scripts permettant de les activer et désactiver sont placés dans le répertoire `/etc/sysconfig/network-scripts/`. Même si certains fichiers d'interface peuvent différer d'un système à l'autre, ce répertoire contient trois types de fichiers:

- *les fichiers de configuration d'interface;*
- *les scripts de contrôle d'interface;*
- *les fichiers de fonctionnement réseau.*

Les fichiers faisant partie de chacune de ces catégories fonctionnent en coopération afin de permettre l'activation de divers périphériques réseau sous Red Hat Linux.

Ce chapitre explore la relation entre ces fichiers et différentes options d'utilisation.

8.1. Fichiers de configuration réseau

Avant d'examiner les fichiers de configuration d'interface eux-mêmes, établissons la liste des fichiers de configuration utilisés pour configurer le réseau. Le fait de comprendre le rôle joué par ces fichiers dans la mise en place de la pile de réseau peut s'avérer utile lors de la personnalisation de votre système Red Hat Linux.

Les fichiers de configuration de réseau primaire sont les suivants:

- `/etc/hosts` — L'objectif principal de ce fichier est de résoudre noms d'hôtes n'ayant pu être résolus d'une autre façon. Il peut également être utilisé pour résoudre des noms d'hôtes sur de petits réseaux ne disposant pas de serveur DNS. Quel que soit le type de réseau utilisé par l'ordinateur, ce fichier doit contenir une ligne spécifiant l'adresse IP du périphérique de bouclage (loopback) (`127.0.0.1`) en tant que `localhost.localdomain`. Pour obtenir davantage d'informations sur ce fichier, consultez la page de manuel relative aux hôtes.
- `/etc/resolv.conf` — Ce fichier précise les adresses IP des serveurs DNS et le domaine de recherche. À moins d'être configuré autrement, les scripts d'initialisation du réseau sont contenus dans ce fichier. Pour obtenir davantage d'informations sur ce fichier, voyez la page de manuel `resolv.conf`.
- `/etc/sysconfig/network` — Précise les informations de routage et d'hébergement (hôte) pour toutes les interfaces de réseau. Pour obtenir davantage d'informations sur ce fichier et sur les directives qu'il accepte, reportez-vous à la Section 4.1.23.
- `/etc/sysconfig/network-scripts/ifcfg-`<*`interface-name`*> — Pour chaque interface réseau sur un système Red Hat Linux, il existe un script de configuration d'interface correspondant. Chacun de ces fichiers fournit des informations spécifiques à une interface réseau particulière. Consultez la Section 8.2 pour obtenir davantage d'informations sur ce type de fichier et les directives qu'il accepte.



Avertissement

Le répertoire `/etc/sysconfig/networking/` est utilisé par l'**Outil d'administration de réseau** (`redhat-config-network`) et son contenu ne doit pas être modifié manuellement. Pour

obtenir davantage d'informations sur la configuration des interfaces de réseau utilisant l'**Outil d'administration de réseau**, voyez le chapitre intitulé *Configuration réseau* du *Guide de personnalisation de Red Hat Linux*.

8.2. Fichiers de configuration d'interface

Les fichiers de configuration d'interface contrôlent le fonctionnement des interfaces logicielles associées aux périphérique réseau individuels. Lorsque votre système Red Hat Linux démarre, il utilise ces fichiers pour savoir quelles interfaces il doit afficher automatiquement et comment les configurer. Ces fichiers sont en général nommés `ifcfg-<nom>`, où `<nom>` se rapporte au nom du périphérique contrôlé par le fichier de configuration.

8.2.1. Interfaces Ethernet

Le fichier `ifcfg-eth0` constitue l'un des fichiers d'interface les plus communs; il contrôle la première *carte d'interface réseau* Ethernet ou *NIC* (de l'anglais 'Network Interface Card') dans le système. Dans un système comportant plusieurs cartes, il y aura plusieurs fichiers `ifcfg-eth<X>` (où <*X*> correspond à un numéro unique associé à une interface spécifique). Étant donné que chaque périphérique a son propre fichier de configuration, un administrateur peut contrôler sur le fonctionnement individuel de chaque interface.

Un fichier `ifcfg-eth0` pour un système utilisant une adresse IP fixe ressemble à ce qui suit:

```
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
NETWORK=10.0.1.0
NETMASK=255.255.255.0
IPADDR=10.0.1.27
USERCTL=no
```

Les valeurs requises dans un fichier de configuration d'interface peuvent changer en fonction d'autres valeurs. Par exemple, le fichier `ifcfg-eth0` pour une interface utilisant DHCP est légèrement différent, car les informations IP sont fournies par le serveur DHCP dans ce cas:

```
DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
```

L'utilitaire **Outil d'administration de réseau** (`redhat-config-network`) permet de modifier facilement les différents fichiers de configuration des interfaces réseau (reportez-vous au chapitre intitulé *Configuration réseau* du *Guide de personnalisation de Red Hat Linux* pour obtenir des instructions détaillées sur l'utilisation de cet outil).

Vous pouvez également modifier manuellement le fichier de configuration pour une interface réseau donnée.

Vous trouverez ci-dessous une liste de paramètres pouvant être configurés dans un fichier de configuration d'interface Ethernet:

- `BOOTPROTO=<protocole>`, où `<protocole>` correspond à l'une des valeurs suivantes:
  - `none` — spécifie qu'aucun protocole de démarrage ne devrait être utilisé.
  - `bootp` — spécifie que le protocole BOOTP devrait être utilisé.
  - `dhcp` — spécifie que le protocole DHCP devrait être utilisé.

- BROADCAST=<*adresse*>, où <*adresse*> correspond à l'adresse de diffusion. Cette directive est déconseillée.
- DEVICE=<*nom*>, où <*nom*> correspond au nom du périphérique physique (à l'exception des périphériques PPP à affectation dynamique où il s'agit du *nom logique*).
- DNS*{1,2}*=<*adresse*>, où <*adresse*> correspond à une adresse de serveur à placer dans /etc/resolv.conf si la directive PEERDNS est réglée sur yes.
- IPADDR=<*adresse*>, où <*adresse*> correspond à l'adresse IP.
- NETMASK=<*masque*>, où <*masque*> correspond à la valeur de masque de réseau.
- NETWORK=<*adresse*>, où <*adresse*> correspond à l'adresse du réseau. Cette directive est déconseillée.
- ONBOOT=<*réponse*>, où <*réponse*> correspond à l'une des valeurs suivantes:
  - yes — spécifie que ce périphérique devrait être activé au démarrage.
  - no — spécifie que ce périphérique ne devrait pas être activé au démarrage.

- PEERDNS=<*réponse*>, où <*réponse*> correspondant à l'une des valeurs suivantes:
  - yes — Modifiez /etc/resolv.conf si la directive DNS est réglée. Si vous utilisez DCHP, yes est la valeur par défaut.
  - no — Ne modifiez pas /etc/resolv.conf.
  - SRCADDR=<*adresse*>, où <*adresse*> correspond à l'adresse IP source spécifiée pour les paquets sortants.
  - USERCTL=<*réponse*>, où <*réponse*> correspondant à l'une des valeurs suivantes:
    - yes — Les utilisateurs autres que le super-utilisateur sont autorisés à contrôler ce périphérique.
    - no — Les utilisateurs autres que le super-utilisateur ne sont pas autorisés à contrôler ce périphérique.

8.2.2. Interfaces de numérotation

Si vous vous connectez à un réseau comme l'Internet par l'intermédiaire d'une connexion commutée PPP, il vous faut un fichier de configuration pour cette interface.

Le nom des fichiers d'interface PPP est attribué selon le format suivant: ifcfg-ppp<*X*> (où <*X*> représente un numéro unique correspondant à une interface spécifique).

Les fichiers de configuration d'interface PPP créé automatiquement lorsque vous utilisez wvdial, l'utilitaire **Outil d'administration de réseau** ou **Kppp** pour créer un compte de numérotation. Le *Guide de démarrage de Red Hat Linux* contient des instructions relatives à l'utilisation de ces outils de connexions par numérotation au moyen d'une GUI. Vous pouvez aussi créer et éditer ce fichier manuellement.

Un fichier ifcfg-ppp0 typique ressemble à l'extrait ci-dessous:

```
DEVICE=ppp0
NAME=test
WVDIALSECT=test
```

```
MODEMPORT=/dev/modem
LINESPEED=115200
PAPNAME=test
USERCTL=true
ONBOOT=no
PERSIST=no
DEFROUTE=yes
PEERDNS=yes
DEMAND=no
IDLETIMEOUT=600
```

Le protocole Internet ligne série (SLIP) (de l'anglais 'Serial Line Internet Protocol') constitue une autre interface de connexion commutée, même s'il est moins fréquemment utilisé. Les fichiers SLIP ont des noms de fichiers de configuration d'interface comme ifcfg-sl0.

Parmi les options dont nous n'avons pas encore parlé, et qui peuvent être utilisées dans ces fichiers, figurent:

- DEFROUTE=<*réponse*>, où <*réponse*> correspond à l'une des valeurs suivantes:
  - yes — spécifie que cette interface doit être configurée comme itinéraire par défaut.
  - no — spécifie que cette interface ne doit pas être configurée comme itinéraire par défaut.
- DEMAND=<*réponse*>, où <*réponse*> correspond à l'une des valeurs suivantes:
  - yes — spécifie que cette interface permettra à pppd d'initialiser une connexion lorsque quelqu'un essaiera de l'utiliser.
  - no — spécifie qu'une connexion doit être établie manuellement pour cette interface.
- IDLETIMEOUT=<*valeur*>, où <*valeur*> correspond au nombre de secondes d'inactivité déclenchant la déconnexion automatique de l'interface.
- INITSTRING=<*chaîne*>, où <*chaîne*> correspond à la chaîne d'initialisation transférée au modem. Cette option est principalement utilisée avec les interfaces SLIP.
- LINESPEED=<*valeur*>, où <*valeur*> correspond à la vitesse de transmission (en bauds) du périphérique. Parmi les valeurs standard possibles figurent 57600, 38400, 19200 et 9600.
- MODEMPORT=<*périphérique*>, où <*périphérique*> correspond au nom du périphérique de série utilisé pour établir la connexion pour l'interface.
- MTU=<*valeur*>, où <*valeur*> correspond au paramètre *unité de transfert maximum (MTU)* (de l'anglais 'Maximum Transfer Unit') pour l'interface. La valeur de MTU correspond au nombre maximal d'octets de données qu'un cadre peut comporter, sans compter les informations en-tête. Dans certaines situations de connexion commutée, si vous réglez ce paramètre à la valeur 576 le nombre de paquets éliminés sera moins important et le débit de connexion sera légèrement amélioré.
- NAME=<*nom*>, où <*nom*> correspond à la référence au titre donné à un ensemble de configurations de connexion commutée.
- PAPNAME=<*nom*>, où <*nom*> correspond au nom d'utilisateur donné durant l'échange de *protocole d'authentification du mot de passe (PAP)* (de l'anglais, 'Password Authentication Protocol') suite auquel vous pouvez vous connecter à un système à distance.
- PEERDNS=<*réponse*>, où <*réponse*> correspond à l'une des valeurs suivantes:
  - yes — spécifie que les entrées /etc/resolv.conf de votre système doivent être modifiées pour utiliser les serveurs DNS fournis par le système à distance lorsqu'une connexion est établie.
  - no — spécifie que le fichier /etc/resolv.conf ne sera pas modifié.

- `PERSIST=`<*réponse*>, où <*réponse*> correspond à l'une des valeurs suivantes:
  - `yes` — spécifie que cette interface doit rester active en permanence, même si elle est désactivée lorsqu'un modem raccroche.
  - `no` — spécifie que cette interface ne doit pas rester active en permanence.

- `REMIP=`<*adresse*>, où <*adresse*> correspond à l'adresse IP du système à distance. Cette valeur n'est en général pas spécifiée.
- `WVDIALSECT=`<*nom*>, où <*nom*> associe cette interface à une configuration de composeur dans `/etc/wvdial.conf`. Ce fichier contient le numéro de téléphone à composer et d'autres informations importantes pour l'interface.

8.2.3. Autres interfaces

Parmi d'autres fichiers de configuration d'interface courants qui utilisent ces options figurent:

- `ifcfg-lo` — une *interface de bouclage* locale (loopback) est souvent utilisée pour effectuer des tests et est aussi utilisée dans un certain nombre d'applications qui nécessitent une adresse IP référant au même système. Toutes les données envoyées au périphérique de bouclage sont immédiatement renvoyées à la couche réseau de l'hôte.

  **Avertissement**

  Ne jamais modifier manuellement le script de l'interface de bouclage, `/etc/sysconfig/network-scripts/ifcfg-lo`. Ce faisant, vous pourriez en effet provoquer un mauvais fonctionnement du système.

- `ifcfg-irlan0` — une *interface infrarouge* permet à des informations de circuler entre des périphériques, tels qu'un ordinateur portable et une imprimante, par l'intermédiaire d'un lien infrarouge qui fonctionne de la même façon qu'un périphérique Ethernet sauf qu'il est généralement utilisé dans une connexion de poste à poste.
- `ifcfg-plip0` — une connexion *Parallel Line Interface Protocol (PLIP)* fonctionne de la même façon, sauf qu'elle utilise un port parallèle.
- `ifcfg-tr0` — les topologies *Token Ring* ne sont pas aussi courantes sur les *Réseaux locaux* (ou *LAN* de l'anglais 'Local Area Networks') qu'elles ne l'étaient autrefois; elles ont été supplantées par Ethernet.

8.2.4. Fichiers alias et clone

Il existe deux types de fichiers de configuration d'interface moins utilisés et se trouvant dans `/etc/sysconfig/network-scripts`: les fichiers *alias* et *clone* qui incluent un composant supplémentaire dans le nom du fichier.

Le nom des fichiers de configuration d'interface alias est attribué selon le format suivant: `ifcfg-`<*if-nom*>`:`<*alias-valeur*>. Ces fichiers permettent à un alias de désigner une interface. Par exemple, un fichier `ifcfg-eth0:0` peut être configuré pour spécifier `DEVICE=eth0:0` et une adresse IP statique de 10.0.0.2, servant donc d'alias pour une interface Ethernet déjà configurée pour recevoir ses informations IP via DHCP dans `ifcfg-eth0`. À ce moment, le périphérique `eth0` est lié à une adresse IP dynamique, mais il est toujours possible d'y faire référence sur ce système via l'adresse IP fixe 10.0.0.2.

Le nom d'un fichier de configuration d'interface clone ressemble à ceci: `ifcfg-`<*if-nom*>`-`<*nom-clone*>. Alors qu'un fichier alias permet de faire référence à un fichier de configuration d'interface existant, un fichier clone permet de spécifier des options complémentaires pour une interface. Par exemple, si vous avez une interface Ethernet DHCP standard appelée `eth0`, le fichier pourrait ressembler à cet extrait:

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp
```

Puisque `USERCTL` est réglé sur `no` si aucune valeur n'est spécifiée, les utilisateurs ne peuvent pas mettre cette interface en fonction ou hors service. Pour permettre aux utilisateurs de le faire, créez un clone en copiant `ifcfg-eth0` dans `ifcfg-eth0-user` puis ajoutez la ligne suivante:

```
USERCTL=yes
```

Lorsqu'un utilisateur met en fonction l'interface `eth0` avec la commande `ifup eth0-user`, les options de configuration de `ifcfg-eth0` sont combinées à celles de `ifcfg-eth0-user`. Ceci n'est qu'un exemple de base, mais cette méthode peut être utilisée avec des options et interfaces diverses.

La méthode la plus simple pour la création de fichiers de configuration d'interface alias et clone consiste à utiliser l'outil graphique, **Outil d'administration de réseau**. Pour en savoir plus sur cet outil, voyez le chapitre intitulé *Configuration réseau* du *Guide de personnalisation de Red Hat Linux*.

8.3. Scripts de contrôle d'interface

Les scripts de contrôle d'interface contrôlent la mise en fonction (activation) et la mise hors service (désactivation) des connexions d'interface. Il existe deux scripts de contrôle principaux, soit `/sbin/ifdown` et `/sbin/ifup`, utilisant d'autres scripts de contrôle situés dans le répertoire `/etc/sysconfig/network-scripts`.

Les scripts d'interface `ifdown` et `ifup` constituent des liens symboliques vers des scripts du répertoire `/sbin/`. Lorsque l'un ou l'autre de ces scripts est appelé, la valeur de l'interface doit être spécifiée, comme par exemple:

```
ifup eth0
Determining IP information for eth0... done.
```

À ce moment, les fichiers `/etc/sysconfig/network-scripts/network-functions` et `/etc/rc.d/init.d/functions` sont approvisionnés et des fonctions de ces fichiers sont utilisées pour diverses tâches. Reportez-vous à la Section 8.4 pour de plus amples informations.

Après avoir vérifié qu'une interface a été spécifiée et que l'utilisateur effectuant la requête est autorisé à activer ou désactiver l'interface, le script correspondant au type de périphérique d'interface est activé. Il s'agit du script qui active et désactive véritablement l'interface. La liste ci-dessous énumère Les scripts de contrôle d'interface les plus courants:

- `ifup-aliases` — configure des alias IP à partir des fichiers de configuration d'interface quand plusieurs adresses IP sont associées à une interface.
- `ifdown-cipcb` et `ifup-cipcb` — permettent d'activer et de désactiver les connexions *Crypto IP Encapsulation (CIPE)* vers le haut et le bas.
- `ifdown-ipv6` et `ifup-ipv6` — contiennent des fonctions associées IPv6 utilisant les variables d'environnement dans divers fichiers de configuration d'interface et `/etc/sysconfig/network`.
- `ifup-ipx` — permet d'activer une interface IPX.

- ifup-plip — permet d'activer une interface PLIP.
- ifup-plusb — permet d'activer une interface USB pour les connexions réseau.
- ifdown-post et ifup-post — contiennent des commandes à exécuter après l'activation ou la désactivation d'une interface spécifique.
- ifdown-ppp et ifup-ppp — permettent d'activer ou de désactiver une interface PPP .
- ifup-routes — ajoute des itinéraires statiques pour un périphérique particulier lorsque son interface est activée.
- ifdown-sit et ifup-sit — contiennent des fonctions associées à l'activation et la désactivation d'un tunnel IPv6 au sein d'une connexion IPv4.
- ifdown-sl et ifup-sl — permettent d'activer ou de désactiver une interface SLIP.



La suppression ou la modification de ces scripts dans le répertoire /etc/sysconfig/network-scripts/ peut provoquer le mauvais fonctionnement ou l'échec de diverses connexions. Seuls les utilisateurs chevronnés peuvent se permettre de modifier les scripts concernant une interface réseau.

Pour simplifier la manipulation simultanée de tous les scripts réseau, utilisez la commande /sbin/service sur le service de réseau (/etc/rc.d/init.d/network), comme ci-dessous:

```
/sbin/servicenetwork<action>
```

Dans cet exemple, <*action*> peut correspondre à start, stop ou restart.

Pour afficher une liste des périphériques configurés et des interfaces réseau actuellement actives, utilisez la commande suivante:

```
/sbin/service/networkstatus
```

8.4. Fichiers de fonctions réseau

Red Hat Linux utilise plusieurs fichiers contenant des fonctions importantes utilisées de diverses façons pour activer et désactiver les interfaces. Plutôt que de forcer chaque fichier de contrôle d'interface à contenir les mêmes fonctions que les autres, ces fonctions sont regroupées dans quelques fichiers utilisés en fonction des besoins.

Le fichier /etc/sysconfig/network-scripts/network-functions contient divers fonctions IPv4 courantes utilisées par bon nombre de scripts de contrôle d'interface. Ces fonctions permettent entre autres, de contacter des programmes en cours d'exécution ayant demandé des informations sur les modifications du statut d'une interface; de configurer des noms d'hôte; de trouver un périphérique passerelle; de vérifier le statut d'un périphérique particulier et d'ajouter un itinéraire par défaut.

Les fonctions requises pour les interfaces IPv6 étant différentes de celles requises pour les interfaces IPv4, un fichier network-functions-ipv6 est spécifiquement conçu pour contenir ces informations. La prise en charge IPv6 doit être activée dans le noyau pour la communication via ce protocole. Une fonction du fichier network-functions vérifie la présence de la prise en charge IPv6. Ce fichier contient également des fonctions permettant de configurer et d'effacer des itinéraires IPv6 statiques, de créer et de supprimer des tunnels, d'ajouter à et de supprimer des adresses IPv6 d'une interface et de rechercher l'existence d'une adresse IPv6 sur une interface.

8.5. Ressources supplémentaires

Les ressources suivantes contiennent davantage d'informations sur les interfaces réseau.

8.5.1. Documentation installée

- `/usr/share/doc/initscripts-<version>/sysconfig.txt` — Un guide complet des options disponibles pour les fichiers de configuration de réseau, y compris les options IPv6 n'ayant pas été abordées dans ce chapitre.
- `/usr/share/doc/iproute-<version>/ip-cref.ps` — Ce fichier Postscript™ contient un grand nombre d'informations sur la commande `ip`, qui peut être utilisée, entre autres, pour manipuler des tables de routage. Utilisez l'application **ghostview** ou **kghostview** pour accéder à ce fichier.





Chapitre 9. Le système de fichiers réseau (NFS)

Le système de fichiers réseau, ou *NFS ('Network File System')*, permet aux hôtes de monter des partitions sur un système distant et de les utiliser exactement comme des systèmes de fichier locaux. Ceci permet à l'administrateur système de stocker des ressources dans un emplacement central du réseau, fournissant ainsi aux utilisateurs légitimes un accès permanent.

Deux versions de NFS sont actuellement en vigueur. La version 2 de NFS (NFSv2) d'une part, utilisée depuis plusieurs années, est largement supportée par divers systèmes d'exploitation. La version 3 de NFS (NFSv3) d'autre part, apporte d'autres fonctions, y compris un traitement de fichiers de taille variable et un meilleur rapportage d'erreurs. Red Hat Linux supporte les deux versions, NFSv2 et NFSv3, et utilise NFSv3 par défaut lors de la connexion à un serveur qui le supporte.

Ce chapitre se concentre sur la version 2 de NFS, mais la plupart des concepts discutés ici s'appliquent aussi à la version 3. De plus, ici ne seront abordés que les concepts fondamentaux de NFS et des références supplémentaires seront indiquées. Pour des instructions spécifiques concernant la configuration et l'exploitation de NFS sur des ordinateurs serveurs ou clients, voyez le chapitre intitulé *Network File System (NFS)* du *Guide de personnalisation de Red Hat Linux*.

9.1. Méthodologie

Linux utilise certes une combinaison de support de niveau noyau et des processus de démons constamment actifs pour fournir le partage de fichier NFS, mais NFS doit toutefois être activé dans le niveau Linux pour fonctionner. NFS utilise les *Appels de procédure distante, ou RPC,* (de l'anglais 'Remote Procedure Calls') pour router des requêtes entre clients et serveurs, ce qui signifie que le service `portmap` doit être activé et en opérationnel aux niveaux d'exécution adéquats pour que la communication NFS aie lieu. De concert avec `portmap`, les processus suivants garantissent qu'une connexion NFS est autorisée et peut continuer sans erreurs:

- `rpc.mountd` — Le processus actif qui reçoit la requête de montage d'un client NFS et vérifie qu'elle correspond bien avec un système de fichiers actuellement exporté.
- `rpc.nfsd` — Le processus qui implémente les composants de l'espace-utilisateur (user-space) du service NFS. Il fonctionne avec le noyau Linux pour satisfaire les requêtes dynamiques des clients NFS, comme par exemple en fournissant des fils de serveur supplémentaires à utiliser par les clients NFS.
- `rpc.lockd` — Le démon qui ne se trouve pas nécessairement dans les noyaux modernes. Le verrouillage de fichier NFS est à présent assuré par le noyau. Il est inclus dans le paquetage `nfs-utils` pour les utilisateurs d'anciens noyaux, qui n'incluent pas cette fonctionnalité par défaut.
- `rpc.statd` — Il implémente le *Moniteur de statut de réseau, ou NSM* (de l'anglais 'Network Status Monitor') de protocole RPC. Ceci fournit une notification de redémarrage lorsqu'un serveur NFS est redémarré sans avoir été éteint correctement.
- `rpc.rquotad` — Le serveur RPC qui fournit des informations de quotas d'utilisateurs pour les utilisateurs distants.

Tous ces programmes ne sont pas indispensables pour le service NFS. Les seuls services devant vraiment être activés sont `rpc.mountd`, `rpc.nfsd` et `portmap`. Les autres démons fournissent des fonctionnalités supplémentaires et ne devraient être utilisés que l'environnement de serveur l'exige.

La version 2 de NFS utilise le *Protocole Datagram d'utilisateur, ou UDP* (de l'anglais 'User Datagram Protocol'), pour fournir une connexion sans déclaration de réseau entre le client et le serveur. La version 3 de NFS peut utiliser UDP ou TCP sur une IP. La connexion sans déclaration UDP réduit le trafic de réseau, puisque le serveur NFS envoie au client un cookie qui l'autorise à accéder au volume

partagé. Ce cookie est une valeur aléatoire stockée du côté serveur et transmis en même temps que les requêtes RPC du client. Non seulement le serveur NFS peut être redémarré sans affecter le client mais le cookie restera intact.

NFS n'effectue d'authentification que lorsqu'un système client tente de monter un système de fichier distant. Pour limiter l'accès, le serveur NFS commence par employer les enveloppeurs TCP. Ceux-ci lisent les fichiers `/etc/hosts.allow` et `/etc/hosts.deny` pour déterminer si un client particulier doit se voir refuser ou accorder l'accès au serveur NFS. Pour plus d'informations sur la configuration des contrôles d'accès avec les enveloppeurs TCP, consultez le Chapitre 15.

Après autorisation d'accès du client permise par les enveloppeurs TCP, le serveur NFS se réfère à son fichier de configuration, `/etc/exports`, pour déterminer si le client peut monter l'un des systèmes de fichier exportés. Après avoir autorisé l'accès, toute opération de fichiers ou de répertoires est envoyée au serveur par le biais d'appels de procédure distante.



Les privilèges de montage NFS sont accordés spécifiquement à un client, pas à un utilisateur. Tout utilisateur d'un ordinateur distant peut accéder aux systèmes de fichiers exportés.

Lorsque vous configurez le fichier `/etc/exports`, soyez très prudent lors de l'attribution des permissions de lecture et écriture (`rw`) à un système de fichiers exporté.

9.1.1. NFS et `portmap`

Pour son fonctionnement, NFS dépend des Appels de procédure distante (ou RPC). Le service `portmap` est nécessaire pour orienter les requêtes RPC vers les services appropriés. Les processus RPC s'annoncent à `portmap` lorsqu'ils démarrent, révélant le numéro de port qu'ils contrôlent et les numéros de programmes RPC qu'ils entendent servir. Le système client contacte alors `portmap` sur le serveur avec un numéro de programme RPC particulier. Le service `portmap` redirige ensuite le client vers le numéro de port correct afin de communiquer avec le service souhaité.

Parce que les services utilisant RPC se basent sur `portmap` pour assurer toutes les connexions avec les requêtes de client entrantes, `portmap` doit être disponible avant le démarrage de chacun de ces services. Si, pour une raison ou pour une autre, le service `portmap` quitte de façon inattendue, redémarrez `portmap` et tous les services actifs au moment de son démarrage.

Le service `portmap` peut s'employer avec les fichiers d'accès d'hôtes des enveloppeurs TCP (`/etc/hosts.allow`et `/etc/hosts.deny`) pour contrôler les systèmes distants qui sont autorisés à utiliser les services basés sur RPC sur le serveur. Reportez-vous au Chapitre 15 pour plus d'informations. Les règles de contrôle d'accès pour `portmap` affecteront tous les services basés sur RPC. Il est également possible de spécifier chacun des démons RPC NFS devant être affectés par une règle de contrôle d'accès particulière. Les pages de manuel relatives à `rpc.mountd` et `rpc.statd` contiennent des informations sur la syntaxe précise de ces règles.

9.1.1.1. Résolution de problèmes liés à NFS avec `portmap`

Puisque `portmap` fournit la coordination entre les services RPC et les numéros des ports utilisés pour communiquer avec eux, il est utile de pouvoir visualiser les services RPC en cours à l'aide de `portmap` lors de la résolution de problèmes. La commande `rpcinfo` montre chaque service basé sur RPC avec son numéro de port, numéro de programme RPC, version, et type de protocole IP (TCP ou UDP).

Pour s'assurer que les bons services NFS basés sur RPC sont activés pour `portmap`, utilisez la commande `rpcinfo -p`:

```
program vers proto   port
```

```
100000    2   tcp    111  portmapper
100000    2   udp    111  portmapper
100024    1   udp   1024  status
100024    1   tcp   1024  status
100011    1   udp    819  rquotad
100011    2   udp    819  rquotad
100005    1   udp   1027  mountd
100005    1   tcp   1106  mountd
100005    2   udp   1027  mountd
100005    2   tcp   1106  mountd
100005    3   udp   1027  mountd
100005    3   tcp   1106  mountd
100003    2   udp   2049  nfs
100003    3   udp   2049  nfs
100021    1   udp   1028  nlockmgr
100021    3   udp   1028  nlockmgr
100021    4   udp   1028  nlockmgr
```

L'option `-p` vérifie le mappeur de ports sur l'hôte spécifié, ou par défaut sur l'hôte local en l'absence de spécification. D'autres options sont décrites dans la page de manuel `rpcinfo`.

Les résultats ci-dessus montrent des services NFS en activité. Si l'un des services NFS services ne démarre pas correctement, `portmap` sera incapable d'orienter les requêtes RPC de clients pour ce service, vers le port adéquat. Souvent, le redémarrage de NFS en étant connecté en tant que super-utilisateur (ou root) (`/sbin/service nfs restart`) permettra à ces services de s'enregistrer correctement auprès de `portmap` et de commencer à fonctionner.

9.2. Les fichiers de configuration du serveur NFS

La configuration d'un système pour le partage des fichiers et répertoires grâce à NFS est très simple. Chaque fichier exporté vers les utilisateurs distants via NFS, ainsi que les droits d'accès liés à ces systèmes de fichiers, se trouvent dans le fichier `/etc/exports`. Ce fichier est lu par la commande `exportfs` pour fournir à `rpc.mountd` et `rpc.nfsd` les informations nécessaires pour le montage distant d'un système de fichier par un hôte autorisé.

La commande `exportfs` permet au super-utilisateur d'exporter ou désexporter sélectivement des répertoires sans redémarrer le service NFS. Lorsque `exportfs` a passé les options appropriées, les systèmes de fichiers à exporter sont écrits dans `/var/lib/nfs/xtab`. Puisque `rpc.mountd` se réfère au fichier `xtab` lorsqu'il décide des privilèges d'accès à un système de fichier, les changements apportés à la liste des systèmes de fichiers exportés prennent effet immédiatement.

Diverses options sont disponibles lorsque l'on utilise `exportfs`:

- `-r` — provoque l'exportation de tous les répertoires listés dans `/etc/exports` par la construction d'une nouvelle liste d'exportation dans `/etc/lib/nfs/xtab`. Cette option rafraîchit effectivement la liste d'exportations par tout changement apporté à `/etc/exports`.
- `-a` — provoque l'exportation ou la désexportation de tous les répertoires, selon les autres options d' `exportfs`.
- `-o` *`options`* — permet à l'utilisateur de spécifier les répertoires à exporter qui ne sont pas listés dans `/etc/exports`. Ces partages de systèmes de fichiers supplémentaires doivent être écrits de la même manière qu'ils sont spécifiés dans `/etc/exports`. Cette option sert à tester un système de fichiers exporté avant de l'ajouter de façon permanente à la liste des systèmes de fichiers à exporter.
- `-i` — ne prend pas en compte `/etc/exports`; seules les options données par la ligne de commande sont utilisées pour définir des systèmes de fichiers exportés.

- -u — désexporte des répertoires de leur montage par des utilisateurs distants. La commande exportfs -ua suspend effectivement le partage de fichier NFS tout en laissant actifs les divers démons NFS. Pour permettre au partage NFS de continuer, tapez exportfs -r.
- -v — opération prolixe selon laquelle les systèmes de fichiers à exporter ou désexporter sont affichés avec de plus amples détails, lors de l'exécution de la commande exportfs.

Si aucune option n'est transmise à la commande exportfs, elle affiche une liste des systèmes de fichiers exportés en cours.

Les changements apportés à /etc/exports peuvent aussi se lire en rechargeant le service NFS à l'aide de la commande service nfs reload. Ce faisant, les démons NFS demeurent actifs tout en réexportant le fichier /etc/exports.

9.2.1. /etc/exports

Le fichier /etc/exports permet non seulement de contrôler quels systèmes de fichiers sont exportés vers des hôtes distants mais permet également de spécifier des options. Les lignes vierges ne sont pas prises en compte, des commentaires peuvent être insérés grâce au symbole dièse (#) et un retour à la ligne peut être introduit grâce à une barre oblique inverse (\). Chaque fichier exporté doit avoir sa ligne propre. Les hôtes autorisés placés après un système de fichiers exporté doivent être séparés par des espaces. Les options pour chacun des hôtes doivent être placées entre parenthèses directement après l'identifieur d'hôte, sans espace entre l'hôte et la première parenthèse.

Dans sa forme la plus simple, /etc/exports requiert seulement le répertoire exporté et l'hôte autorisé à l'exploiter:

```
/some/directory bob.example.com
/another/exported/directory 192.168.0.3
```

Après la réexportation de /etc/exports grâce à la commande /sbin/service nfs reload, l'hôte bob.example.com pourra monter /some/directory et 192.168.0.3 peut monter /another/exported/directory. Parce qu'aucune option n'est spécifiée dans cet exemple, plusieurs préférences par défaut de NFS entrent en vigueur:

- ro — Lecture-seule. Les hôtes qui montent ce système de fichiers ne pourront pas le modifier. Pour autoriser les hôtes à apporter des modifications au système de fichiers, l'option rw (Lecture-écriture) doit être spécifiée.
- async — Permet au serveur d'écrire des données sur le disque lorsqu'il le juge opportun. Bien que cela ne soit pas important si l'hôte reçoit des données en lecture-seule, s'il apporte des modifications à un système de fichiers en lecture-écriture et que le serveur plante, des données peuvent être perdues. En spécifiant l'option sync, toutes les opérations d'écriture doivent être confiées au disque avant que la requête d'écriture par le client ne soit effectivement achevée. Cela peut diminuer les performances.
- wdelay — Cette option entraîne un retard des opération décriture sur le disque par NFS, s'il suspecte qu'une autre requête d'écriture est imminente. Ce faisant, les performances peuvent être améliorées grâce à une réduction du nombre d'accès au disque par des commandes d'écriture séparées, réduisant ainsi le temps d'écriture. L'option no_wdelay quant à elle, désactive cette fonction mais n'est disponible que lors de l'utilisation de l'option sync.
- root_squash — Retire au super-utilisateur en connexion distante tous les privilèges de son status en lui assignant l'ID d'utilisateur 'personne'. Ce faisant, le pouvoir du super-utilisateur distant est réduit au niveau d'utilisateur le plus bas, lui empêchant d'agir comme comme s'il était le super-utilisateur sur le système local. Sinon, l'option no_root_squash annule cette fonction de réduction des privilèges du super-utilisateur. Afin de limiter le champ d'action de chaque utilisateur distant, y compris le super-utilisateur, utilisez l'option all_squash. Pour spécifier l'utilisateur et ID de groupe à utiliser avec des utilisateurs distants d'un hôte particulier,

utilisez respectivement les options `anonuid` et `anongid`. Dans ce cas, vous pouvez créer un compte d'utilisateur spécial pour que les utilisateurs NFS distants partagent et spécifient, `(anonuid=`<*uid-valeur*>`,anongid=`<*valeur-gid*>`)`, où <*valeur-uid*> correspond au numéro de l'ID d'utilisateur et <*gid-valeur*> représente le numéro de l'ID de groupe.

Pour outrepasser ces réglages par défaut, vous devez spécifier une option qui les remplace. Par exemple, si vous ne spécifiez pas `rw`, cette exportation sera partagée seulement en lecture. Ce remplacement des réglages par défaut doit être explicitement spécifié pour chaque système de fichier exporté. De plus, d'autres options sont disponibles là où il n'existe pas de valeur par défaut. Elles permettent d'annuler la vérification de sous-arborescence, l'accès à des ports non-sûrs et les verrouillages non-sûrs de fichiers (nécessaires pour certaines implémentations anciennes de client NFS). Consultez la page de manuel relative à `exports` pour plus de détails sur ces options moins souvent utilisées.

Lors de la précisions des nom d'hôtes, utilisez les méthodes suivantes:

- *hôte simple* (single host) — Où un hôte particulier est spécifié avec un nom de domaine, d'hôte ou une adresse IP pleinement qualifiés.
- *caractères génériques* (wildcards) — Où les caractères `*` ou `?` sont utilisés pour prendre en compte un groupement de noms de domaines ou adresses IP pleinement qualifiés ou ceux qui correspondent à une particulière chaîne de lettres.

  Soyez toutefois prudents si vous utilisez des caractères génériques pour des noms de domaines pleinement qualifiés, car ils sont souvent plus exacts que ce que vous escomptiez. Par exemple, si vous utilisez `*.example.com` comme caractère générique, sales.example.com sera autorisé à accéder au système de fichiers exporté, mais bob.sales.example.com lui, ne le sera pas. Pour retenir les deux noms de domaine, ainsi que sam.corp.example.com, vous devrez utiliser `*.example.com *.*.example.com`.
- *réseaux IP* (IP networks) — Autorise la mise en correspondance d'hôtes selon leur adresse IP dans un réseau plus grand. Par exemple, `192.168.0.0/28` autorisera les 16 premières adresses IP, de 192.168.0.0 à 192.168.0.15, à accéder au système de fichiers exporté, mais pas 192.168.0.16 ou une adresse IP supérieure.
- *groupes réseau* (netgroups) — Attribue un nom de groupe à un groupe réseau NIS, écrit ainsi: `@`<*nom-du-groupe*>. Cette option attribue au serveur NIS la charge du contrôle d'accès pour ce système de fichier exporté, où les utilisateurs peuvent être ajoutés et supprimés dans un groupe NIS sans affecter `/etc/exports`.



Avertissement

Le format du fichier `/etc/exports` est très précis, particulièrement en ce qui concerne l'utilisation des caractères d'espace. Rappelez-vous bien de toujours séparer les systèmes de fichiers exportés des hôtes, et les hôtes entre eux à l'aide d'un caractère d'espace. Toutefois, aucun autre caractère d'espace ne doit se trouver dans le fichier à moins qu'ils ne soient utilisés pour des lignes de commentaire.

Par exemple, les deux lignes suivantes n'ont pas la même signification:

```
/home bob.example.com(rw)
/home bob.example.com (rw)
```

La première ligne autorise seulement les utilisateurs de `bob.example.com` à avoir une accès en lecture-écriture au répertoire `/home`. La deuxième ligne elle, autorise les utilisateurs de `bob.example.com` à monter le répertoire en lecture-seule (valeur par défaut), mais tout autre utilisateur peut le monter en lecture-écriture.

9.3. Les fichiers de configuration de clients NFS

Tout partage NFS proposé par un serveur peut être monté à l'aide de diverses méthodes. Le partage peut bien sûr être monté manuellement, en utilisant la commande `mount`. Pour ce faire, le super-utilisateur doit taper la commande `mount` chaque fois que le système redémarre. Parmi les deux méthodes de configuration de montage NFS figurent: la modification de `/etc/fstab` ou l'utilisation du service `autofs`.

9.3.1. `/etc/fstab`

L'insertion d'une ligne correctement formatée dans le fichier `/etc/fstab` revient à monter manuellement le système de fichiers exporté. Le fichier `/etc/fstab` est lu par le script `/etc/rc.d/init.d/netfs` au démarrage du système et tout partage spécifié dans ce dernier sera monté.

Une ligne `/etc/fstab` courante pour monter une exportation NFS ressemblera à ceci:

```
<serveur>:</path/of/dir> </local/mnt/point> nfs <options> 0 0
```

L'option <*serveur-hôte*> correspond au nom d'hôte, l'adresse IP ou le nom de domaine pleinement qualifié du serveur exportant le système de fichiers.

L'option <*/path/of/directory*> correspond au chemin vers le répertoire exporté.

L'option <*/local/mount/point*> spécifie l'endroit dans le système de fichier local où monter le répertoire exporté. Ce point de montage doit être déterminé avant que `/etc/fstab` ne soit lu, sinon le montage échouera.

L'option `nfs` spécifie le type de système de fichiers en cours de montage.

La zone <*options*> spécifie les options de montage pour le système de fichiers. Par exemple, si la zone d'options stipule `rw,suid`, le système de fichier exporté sera monté en lecture-écriture (read-write) les ID d'utilisateur et de groupe fixées par le serveur seront utilisées. Notez que les parenthèses ne doivent pas être utilisées ici. Pour obtenir des informations sur des options de montage supplémentaires, consultez la Section 9.3.3.

9.3.2. `autofs`

Un inconvénient lors de l'utilisation de `/etc/fstab` est que, indépendamment de la fréquence d'utilisation de ce système de fichiers monté, votre système doit allouer des ressources pour conserver ce montage en place. Ceci n'est pas un problème pour un ou deux montages, mais si votre système maintient le montage de douzaines de systèmes à la fois, les performances générales du système peuvent en pâtir. Une alternative à `/etc/fstab` consiste à utiliser l'utilitaire basé sur le noyau nommé `automount`, qui lui montera et démontera les systèmes de fichiers NFS automatiquement, économisant ainsi des ressources.

Le script `autofs`, situé dans le répertoire `/etc/rc.d/init.d/`, sert à contrôler `automount` par le biais du fichier de configuration primaire `/etc/auto.master`. Alors qu'`automount` peut être spécifié dans une ligne de commande, il est plus commode de spécifier les points de montage, nom d'hôte, répertoire exporté et options dans un ensemble de fichiers, plutôt que de les taper tous à la main. En exécutant `autofs` en tant qu'un service démarrant et arrêtant les niveaux d'exécution spécifiés, les configurations de montage des divers fichiers peuvent être automatiquement implémentées.

Les fichiers de configuration `autofs` sont organisés selon une relation parent-enfant. Un fichier de configuration principal (`/etc/auto.master`) se réfère à des points de montage sur votre système qui sont liés à un *type de correspondance* (map type) particulier, qui prend la forme d'autres fichiers de configuration, programmes, chemins NIS et autres méthodes de montage moins courantes. Le fichier `auto.master` contient des lignes se référant à chacun de ces points de montage, organisées de la manière suivante:

```
<point-de-montage>
<map-type>
```

L'élément <*point-de-montage*> de cette ligne indique l'emplacement du montage sur le système de fichiers local. L'option <*map-type*> fait référence à la manière dont le point de montage sera monté. La méthode la plus courante pour monter automatiquement des exportations NFS consiste à utiliser un fichier en tant que type de chemin (map-type) pour un point de montage particulier. Le fichier de chemin (map file), généralement nommé `auto.`<*point-de-montage*>, où <*point-de-montage*> est le point de montage désigné dans `auto.master`, contient des lignes similaires à celles reproduites ci-dessous:

```
<répertoire>
<options-de-montage>
<hôte>:<système-de-fichiers-exporté>
```

L'élément <*répertoire*> réfère au répertoire dans le point de montage où le système de fichiers exporté devrait être monté. Tout comme une commande `mount` standard, l'hôte exportant le système de fichiers ainsi que le système de fichiers exporté doivent être spécifiés dans la section <*hôte*>:<*système-de-fichiers-exporté*>. Pour spécifier des options particulières pour le montage du système de fichiers exporté, placez-les dans la section <*options-de-montage*>, en les séparant bien par des virgules. Pour des montages NFS utilisant `autofs`, placez `-fstype=nfs` dans la section <*options-de-montage*>.

Bien que les fichiers de configuration `autofs` puissent être utilisés pour divers montages pour divers types de périphériques et systèmes de fichiers, ils se révèlent particulièrement utiles lors de la créaton de montages NFS. Par exemple, des organisations stockent le répertoire `/home/` d'utilisateur sur un serveur central via le partage NFS. Ensuite, elles configurent le fichier `auto.master` sur chacune des stations de travail pour renvoyer à un fichier `auto.home` contenant les spécifications du montage du répertoire `/home/` via NFS. Cela permet à l'utilisateur d'accéder à ses données personnelles et aux fichiers de configuration dans son répertoire `/home/` en se connectant sur n'importe quel ordinateur du réseau interne. Le fichier `auto.master` dans cette situation ressemblerait à l'extrait suivant:

```
/home   /etc/auto.home
```

Ceci installe le point de montage `/home/` sur le système de fichier local à configurer avec le fichier `/etc/auto.home`, qui pourrait ressembler à l'extrait suivant:

```
*  -fstype=nfs,soft,intr,rsize=8192,wsize=8192,nosuid  server.example.com:/home
```

Cette ligne déclare que tout répertoire auquel un utilisateur tente d'accéder dans le répertoire `/home/` local (en raison de l'astérisque) devrait entraîner un montage NFS sur le système server.domain.com au sein de son système de fichiers `/home/` exporté. Les options de montage spécifient que chaque montage NFS de répertoire `/home/` devrait utiliser une suite particulière de paramètres. Pour de plus amples informations sur les options de montage, y compris celles utilisées dans cet exemple, consultez la Section 9.3.3.

9.3.3. Options courantes de montage NFS

Au-delà du montage d'un système de fichiers sur un hôte distant via NFS, un certain nombre d'autres options peuvent être spécifiées au moment du montage, pour le rendre plus commode à utiliser. Ces options peuvent être utilisées avec les commandes manuelles `mount`, les paramètres `/etc/fstab` et `autofs` et d'autres méthodes de montage.

Ci-dessous figurent les options les plus courantes pour les montages NFS:

- `hard` ou `soft` — Spécifie si le programme utilisant un fichier via une connexion NFS doit s'arrêter et attendre (`hard`) que le serveur revienne en ligne si l'hôte servant le système de fichiers exporté est indisponible, ou s'il doit rapporter une erreur (`soft`).

  Si l'option `hard` est spécifiée, l'utilisateur ne peut pas terminer le processus attendant la communication NFS pour recommencer, à moins que l'option `intr` ne soit également spécifiée.

  Si l'option `soft`, est spécifiée, l'utilisateur peut ajouter une option `timeo=`<*valeur*> où <*valeur*> spécifie la durée d'attente en secondes avant de rapporter l'erreur.

- `intr` — Autorise l'interruption des requêtes NFS si le serveur est en panne ou ne peut pas être atteint.
- `nolock` — Peut être nécessaire afin de pouvoir se connecter à d'anciens serveurs NFS. Pour effectuer le verrouillage, utilisez l'option `lock`.
- `noexec` — Interdit l'exécution de binaires sur le système de fichiers monté. Cette option est utile si votre système Red Hat Linux est en train de monter un système de fichiers non-Linux via NFS, contenant des binaires incompatibles.
- `nosuid` — Interdit aux bits identifieur-d'utilisateur-fixé ou identifieur-de-groupe-fixé de prendre effet.
- `rsize=8192` et `wsize=8192` — Peuvent accélérer la communication NFS pour la lecture (`rsize`) et l'écriture (`wsize`) en déterminant une taille de blocs de données supérieure, exprimée en bits, pour les transférer en une fois. Soyez prudent si vous changez ces valeurs; des noyaux Linux ou des cartes de réseau anciens pourraient ne pas fonctionner correctement avec des tailles de blocs supérieures.
- `nfsvers=2` or `nfsvers=3` — Spécifie la version du protocole NFS à utiliser.

La page de manuel relative à `mount` énumère davantage d'options, y compris les options pour monter des systèmes de fichiers autres que les systèmes de fichiers NFS.

9.4. Sécuriser NFS

NFS fonctionne bien pour le partage de systèmes de fichiers entiers avec un grand nombre d'hôtes connus et d'une manière largement transparente. Beaucoup d'utilisateurs accédant aux fichiers grâce à un montage NFS ne se rendent pas compte que le système de fichiers qu'ils sont en train d'utiliser ne se trouve pas vraiment sur leur système local. De ce fait, et avec l'habitude d'utilisation, divers problèmes potentiels de sécurité peuvent surgir.

Les points suivants doivent être considérés lorsque des systèmes de fichiers NFS sont exportés sur un serveur où lorsqu'ils sont montés sur un client. Ce faisant, les risques de sécurité NFS seront minimisés et les données stockées sur le serveur seront mieux protégées.

9.4.1. Accès des hôtes

NFS contrôle qui peut monter un système de fichiers exporté en se basant sur l'hôte qui effectue la requête de montage et non pas sur l'utilisateur qui exploitera effectivement le système de fichiers. Les hôtes doivent se voir accorder des droits explicites pour pouvoir monter le système de fichiers exporté. Le contrôle d'accès n'est possible pour les utilisateurs, que par les permissions de fichier et de répertoire. En d'autres termes, une fois qu'un système de fichiers est exporté via NFS, tout hôte distant connecté au serveur NFS peut avoir accès aux données partagées. Afin de limiter les risques potentiels, les administrateurs système peuvent restreindre l'accès à une lecture-seule ou peuvent réduire les utilisateurs à une ID d'utilisateur et de groupe commune. Ceci étant, de telles solutions peuvent empêcher l'utilisation du partage NFS de la manière originellement prévue.

De plus, si un agresseur prend le contrôle du serveur DNS utilisé par le système effectuant l'exportation du système de fichiers NFS, le système associé avec un nom d'hôte particulier ou

un nom de domaine pleinement qualifié peut renvoyer vers un ordinateur non-légitime. À ce stade, l'ordinateur non-autorisé *devient* le système ayant l'autorisation de monter le partage NFS, puisqu'aucun nom d'utilisateur ou mot de passe n'est échangé pour fournir une sécurité supplémentaire au montage NFS. Les serveurs NIS compromis courent le même risque, si des groupes réseau NIS sont utilisés pour permettre à certains hôtes de monter un partage NFS. En utilisant des adresses IP situées dans `/etc/exports`, ce genre d'attaque devient plus difficile.

Les caractères génériques doivent être utilisés avec parcimonie lorsque la persmission d'exporter des partages NFS est attribuée car le champs d'action de ces caractères génériques peut s'étendre à un plus grand nombre de systèmes que prévus.

Pour de plus amples informatons sur la sécurisation de NFS, reportez-vous au chapitre intitulé *Sécurité du serveur* du *Guide de sécurité de Red Hat Linux*.

9.4.2. Permissions de fichiers

Une fois que le système de fichier NFS est monté en lecture-écriture par un hôte distant, la seule protection dont dispose chacun des fichiers partagés réside dans ses permissions. Si deux utilisateurs partageant la même valeur d'ID d'utilisateur montent le même système de fichier NFS, ils pourront modifier les fichiers mutuellement. De plus, toute personne connectée en tant que super-utilisateur (ou root) sur le système client peut utiliser la commande `su -` pour devenir un utilisateur ayant accès à des fichiers particuliers via un partage NFS. Pour de plus amples informations sur les conflits entre NFS et les ID d'utilisateur, reportez-vous au chapitre intitulé *Gestion de comptes et groupes* du *Guide d'administration système de Red Hat Linux*.

Le comportement par défaut lors de l'exportation d'un système de fichiers via NFS consiste à utiliser la fonction de *réduction du super-utilisateur (ou 'root squashing')*. Cette dernière permet d'assigner à l'ID d'utilisateur d'une personne quelconque accédant au partage NFS en tant que super-utilisateur (ou root) sur son ordinateur local, une valeur du compte personne (nobody) du serveur. Il est vivement conseillé de ne jamais désactiver la fonction de 'root squashing'.

Si l'exportation d'un partage NFS ne doit se faire qu'en lecture-seule, songez à utiliser l'option `all_squash`, qui attribue à tout utilisateur accédant au système de fichiers exporté, l'ID d'utilisateur personne (nobody).

9.5. Ressources supplémentaires

L'administration d'un serveur NFS peut se transformer en un véritable défi. Maintes options, y compris un certains nombre passé sous silence dans ce chapitre, sont disponibles pour l'exportation ou le montage de partages NFS. Pour de plus amples informations, consultez les sources d'information mentionnées ci-dessous.

9.5.1. Documentation installée

- `/usr/share/doc/nfs-utils-`<*numéro-de-version*>`/` — Remplacez <*numéro-de-version*> par le numéro de version du paquetage NFS. Ce répertoire contient de nombreuses informations sur l'implémentation de NFS sous Linux, y compris diverses configurations NFS et leur impact sur les performances de transfert de fichiers.
- `man mount` — Contient une vue complète des options de montage pour les configurations aussi bien de serveur que de client NFS.
- `man fstab` — Donne des détails quant au format du fichier `/etc/fstab` utilisé pour monter les systèmes de fichiers lors du démarrage du système.

- `man nfs` — Fournit des détails non seulement sur l'exportation de systèmes de fichiers spécifique à NFS, mais également sur les options de montage.
- `man exports` — Montre les options couramment utilisées dans le fichier `/etc/exports` lors de l'exportation de systèmes de fichiers NFS.

9.5.2. Livres sur le sujet

- *Managing NFS and NIS* de Hal Stern, Mike Eisler, et Ricardo Labiaga; O'Reilly & Associates — Constitue un excellent guide de référence pour les nombreuses exportations NFS et options de montage disponibles.
- *NFS Illustrated* de Brent Callaghan; Addison-Wesley Publishing Company — Fournit des comparaisons de NFS avec d'autres systèmes de fichiers réseau et montre, en détail, comment se déroule une communication NFS.



Chapitre 10.
Serveur HTTP Apache

Le Serveur HTTP Apache est un serveur Web Open Source robuste de niveau commercial qui a été développé par Apache Software Foundation (http://www.apache.org). Red Hat Linux comprend le Serveur HTTP Apache version 2.0 ainsi que de nombreux modules de serveur conçus pour améliorer sa fonctionnalité.

Le fichier de configuration par défaut installé avec le Serveur HTTP Apache fonctionne dans la plupart des situations sans devoir être modifié. Toutefois, ce chapitre décrit brièvement de nombreux fichiers de configuration du Serveur HTTP Apache (`/etc/httpd/conf/httpd.conf`) pour aider les utilisateurs ayant nécessitant une configuration personnalisée ou devant convertir un fichier de configuration dans l'ancien format 1.3 du Serveur HTTP Apache.



Si vous utilisez l'outil graphique **Outil de configuration HTTP** (`redhat-config-httpd`), *n'éditez pas* manuellement fichier de configuration du Serveur HTTP Apache car l'**Outil de configuration HTTP** crée une nouvelle version de ce fichier chaque fois qu'il est utilisé.

Pour obtenir plus d'informations concernant **Outil de configuration HTTP**, consultez le chapitre intitulé *Configuration du Serveur HTTP Apache* du *Guide de personnalisation de Red Hat Linux*.

10.1. Serveur HTTP Apache 2.0

Il existe des différences importantes entre la version 2.0 et la version 1.3 du Serveur HTTP Apache (version 1.3 fournie avec la version 7.3 de Red Hat Linux et les versions précédentes). Cette section passe en revue quelques-unes des nouvelles fonctions du Serveur HTTP Apache 2.0 et présente les changements importants. Pour obtenir des informations sur la migration d'un fichier de configuration version 1.3 vers le format 2.0, reportez-vous à la Section 10.2.

10.1.1. Fonctions du Serveur HTTP Apache 2.0

Le Serveur HTTP Apache 2.0 apporte bon nombre de nouvelles fonctions, parmi lesquelles:

- *Nouvelle API Apache* — Les modules utilisent un nouvel ensemble d'Interfaces de programmation d'applications (ou API de l'anglais 'Application Programming Interfaces').

  Les modules élaborés pour le Serveur HTTP Apache 1.3 ne fonctionneront pas s'ils ne sont pas portés vers la nouvelle API. Si vous ne savez pas si un module particulier a été porté ou non, consultez l'assistance du paquetage *avant* la mise à niveau.

- *Filtrage* — Les modules peuvent jouer le rôle de filtres de contenu. Reportez-vous à la Section 10.2.4 pour en savoir plus sur le fonctionnement du filtrage.
- *Prise en charge IPv6* — L'adressage IP de nouvelle génération est désormais pris en charge.
- *Directives simplifiées* — Bon nombre de directives complexes ont été supprimées, et d'autres ont été simplifiées. Reportez-vous à la Section 10.5 pour plus d'informations sur les directives spécifiques.

- *Réponses multilingues aux erreurs* — Lors de l'utilisation de documents *'Server Side Include'* (*SSI*), des pages de réponse en cas d'erreur personnalisées peuvent être proposées dans plusieurs langues.
- *Prise en charge multi-protocoles* — De nombreux protocoles sont pris en charge.

Vous trouverez une liste plus complète des changements à l'adresse http://httpd.apache.org/docs-2.0/.

10.1.2. Changements de paquetage dans le Serveur HTTP Apache 2.0

Depuis la version 8.0 de Red Hat Linux, les paquetages du Serveur HTTP Apache a été renommé. Certains paquetages associés ont également été renommés, retirés ou incorporés dans d'autres paquetages.

Vous trouverez ci-dessous une liste des changements de paquetage:

- Les paquetages `apache`, `apache-devil` et `apache-manual` ont été renommés respectivement `httpd`, `httpd-devel` et `httpd-manual`.
- Le paquetage `mod_dav` a été incorporé dans le paquetage `httpd`.
- Les paquetages `mod_put` et `mod_roaming` ont été supprimés car leur fonctionnalité correspond en fait à un sous-ensemble de celle fournie par `mod_dav`.
- Les paquetages `mod_auth_any` et `mod_bandwidth` ont été supprimés.
- Le numéro de version du paquetage `mod_ssl` est désormais synchronisé avec le paquetage `httpd`. Cela signifie que le paquetage `mod_ssl` du Serveur HTTP Apache 2.0 a un numéro de version *plus bas* que le paquetage `mod_ssl` pour le Serveur HTTP Apache 1.3.

10.1.3. Changements du système de fichiers de la version 2.0 du Serveur HTTP Apache

Lorsque vous passez à la version 2.0 du Serveur HTTP Apache, voici les changements apportés au système de fichiers:

- *Un nouveau répertoire de configuration, `/etc/httpd/conf.d/`, a été ajouté.* — Ce nouveau répertoire sert à stocker les fichiers de configuration des modules en paquetages individuels, tels que `mod_ssl`, `mod_perl` et `php`. La directive `Include conf.d/*.conf` demande au serveur de charger les fichiers de configuration à partir de cet emplacement au sein du fichier de configuration du Serveur HTTP Apache, `/etc/httpd/conf/httpd.conf`.

  
  Important

  Lors de la migration d'une configuration existante, cette ligne doit être insérée.

- *Les programmes `ab` et `logresolve` ont été déplacés.* — Ces utilitaires sont passés du répertoire `/usr/sbin/` au répertoire `/usr/bin/`. Par conséquent, les scripts disposant de chemins d'accès absolus pour ces binaires échoueront.
- *La commande `dbmmanage` a été remplacée.* — La commande `dbmmanage` a été remplacée par `htdbm`. Reportez-vous à la Section 10.2.4.4 pour de plus amples informations.
- *Le fichier de configuration `logrotate` à été renommé.* — Le nom du fichier de configuration `logrotate` a été changé de `/etc/logrotate.d/apache` à `/etc/logrotate.d/httpd`.

La section qui suit présente la migration d'une configuration du Serveur HTTP Apache version 1.3 au nouveau format 2.0.

10.2. Migration de fichiers de configuration du Serveur HTTP Apache version 1.3

Si vous effectuez une mise à niveau de la version 7.3 de Red Hat Linux ou d'une version précédente, sur laquelle le Serveur HTTP Apache était déjà installé, le nouveau fichier de configuration pour le paquetage du Serveur HTTP Apache 2.0 sera installé sous `/etc/httpd/conf/httpd.conf.rpmnew` et la version originale 1.3 `httpd.conf` ne sera pas modifiée. Bien sûr, il vous appartient entièrement de décider d'utiliser la nouvelle configuration et de migrer vos anciens paramètres vers celle-ci, ou d'utiliser le fichier existant comme base et de le modifier en fonction de vos besoins; cependant, certaines parties du fichier ayant été plus modifiées que d'autres, une approche mixte est généralement préférable. Les fichiers de configuration pour les versions 1.3 et 2.0 sont divisés en trois sections. L'objectif de ce guide est de suggérer la meilleure procédure à suivre.

Si le fichier `/etc/httpd/conf/httpd.conf` est une version modifiée de la version par défaut Red Hat Linux et qu'une copie sauvegardée de l'original est disponible, le plus simple serait d'appeler la commande `diff` comme dans l'exemple suivant:

```
diff -u httpd.conf.orig httpd.conf | less
```

Cette commande soulignera toutes les modifications apportées. Si une copie du fichier original n'est pas disponible, vous devez l'extraire du paquetage RPM en utilisant les commandes `rpm2cpio` et `cpio`, comme dans l'exemple suivant:

```
   rpm2cpio apache-<version-number>.i386.rpm | cpio -i --make
```

Dans la commande ci-dessous, remplacez *<numéro-version>* par le numéro de version du paquetage `apache`.

Enfin, il est utile de savoir que le Serveur HTTP Apache dispose d'un mode test qui permet de trouver les erreurs de configuration. Pour y accéder, entrez la commande suivante:

```
apachectl configtest
```

10.2.1. Configuration de l'environnement global

La section 'Environnement global' du fichier de configuration contient des directives qui modifient tout le fonctionnement du Serveur HTTP Apache, comme par exemple, le nombre de requêtes simultanées qu'il peut traiter et les emplacements des divers fichiers qu'il utilise. Étant donné que cette section nécessite un grand nombre de changements, comparé aux autres, il est vivement recommandé de baser cette section sur le fichier de configuration du Serveur HTTP Apache version 2.0 et d'y incorporer ensuite les anciens paramètres.

10.2.1.1. Sélection des interfaces et ports à relier

Les directives `BindAddress` et `Port` n'existent plus; leur fonctionnalité est désormais fournie par une directive plus flexible nommée `Listen`.

Si vous aviez mis `Port 80` dans votre fichier de configuration version 1.3, vous devrez le remplacer par `Listen 80` dans le fichier de configuration version 2.0. Si `Port` avait une valeur *autre que 80*, vous devez ajouter le numéro du port au contenu de la directive `ServerName`.

L'extrait ci-dessous, représente un exemple de la directive du Serveur HTTP Apache version 1.3:

```
Port 123
ServerName www.example.com
```

Pour migrer ce paramètre vers la version 2.3 du Serveur HTTP Apache, utilisez la structure suivante:

```
Listen 123
```

```
ServerName www.example.com:123
```

Pour plus d'informations sur le sujet, reportez-vous à la documentation suivante sur le site Web d'Apache Software Foundation:

- `http://httpd.apache.org/docs-2.0/mod/mpm_common.html#listen`
- `http://httpd.apache.org/docs-2.0/mod/core.html#servername`

10.2.1.2. Régulation de la taille du Server-pool

Dans la version 2.0 du Serveur HTTP Apache, la responsabilité de l'autorisation des demandes et l'envoi des processus fils pour les traiter a été synthétisé dans un groupe de modules multi-tâches nommés *'Multi-Processing Modules'* (ou *MPM*). Contrairement à d'autres modules, seul un module du groupe MPM peut être chargé par le Serveur HTTP Apache. Trois modules MPM sont chargés dans la version 2.0, à savoir, `prefork`, `worker` et `perchild`.

Le comportement original du Serveur HTTP Apache version 1.3 a été déplacé dans le MPM `prefork`. Actuellement, seul le MPM `prefork` est disponible sur Red Hat Linux, bien que les autres MPM puissent être disponibles à une date ultérieure.

Étant donné que le MPM `prefork` accepte les mêmes directives que le Serveur HTTP Apache version 1.3, il est possible de transférer les directives suivantes:

- `StartServers`
- `MinSpareServers`
- `MaxSpareServers`
- `MaxClients`
- `MaxRequestsPerChild`

Pour plus d'informations sur le sujet, reportez-vous à la documentation suivante sur le site Web d'Apache Software Foundation:

- http://httpd.apache.org/docs-2.0/mpm.html

10.2.1.3. Prise en charge de DSO ('Dynamic Shared Object')

Un grand nombre de changements étant ici requis, il est vivement recommandé que quiconque essayant de modifier une configuration du Serveur HTTP Apache version 1.3 pour l'adapter à la version 2.0 (par opposition à la migration de vos changements vers la configuration, version 2.0) copie cette section du fichier de configuration Red Hat Linux Serveur HTTP Apache 2.0.

Les utilisateurs ne souhaitant pas copier la section de la configuration du Serveur HTTP Apache version 2.0 devraient prendre note des informations suivantes:

- Les directives `AddModule` et `ClearModuleList` n'existent plus. Elles étaient utilisées pour assurer l'activation des modules dans la bon ordre. L'API du Serveur HTTP Apache 2.0 API permet aux modules de préciser leur d'activation, éliminant ainsi la raison-d'être de ces deux directives.
- L'ordre des lignes `LoadModule` ne se justifie plus.
- De nombreux modules ont été ajoutés, supprimés, renommés, divisés ou incorporés les uns aux autres.

- Les lignes `LoadModule` des modules intégrés dans leurs propres RPM (`mod_ssl`, `php`, `mod_perl`, et similaires) ne sont plus nécessaires puisque vous pouvez les trouver dans le fichier approprié dans le répertoire `/etc/httpd/conf.d/`.
- Les diverses définitions `HAVE_XXX` ne sont plus définies.



Important

Si vous décidez de modifier votre fichier original, notez qu'il est essentiel que le fichier `httpd.conf` contienne la directive suivante:

```
Include conf.d/*.conf
```

L'oubli de cette directive entraînerait l'échec de tous les modules contenus dans leurs propres RPM (tels que `mod_perl`, `php`, et `mod_ssl`).

10.2.1.4. Autres changements de l'environnement global

Les directives suivantes ont été supprimées de la configuration du Serveur HTTP Apache 2.0:

- *`ServerType`* — le Serveur HTTP Apache ne peut être exécuté qu'en tant que `ServerType standalone`, rendant ainsi cette directive inutile.
- *`AccessConfig`* et *`ResourceConfig`* — Ces directives ont été supprimées puisqu'elles reflétaient la fonctionnalité de la directive `Include`. Si les directives `AccessConfig` et `ResourceConfig` sont paramétrées, vous devez les remplacer par des directives `Include`.

  Pour obtenir l'assurance que les fichiers seront lus dans l'ordre désigné par les anciennes directives, vous devrez placer les directives `Include` à la fin du fichier `httpd.conf`, en prenant bien soin placer celui correspondant à `ResourceConfig` avant celui correspondant à `AccessConfig`. Si vous utilisez les valeurs par défaut, vous devez les inclure explicitement dans les fichiers `conf/srm.conf` et `conf/access.conf`.

10.2.2. Configuration du serveur principal

La section de la configuration du serveur principal du fichier de configuration installe le serveur principal, ce qui répond à toute requête non-traitée par une définition `<VirtualHost>`. Des valeurs par défaut sont aussi fournies ici pour tous les fichiers conteneurs `<VirtualHost>` définis.

Les directives utilisées dans cette section ont été légèrement modifiées entre le Serveur HTTP Apache 1.3 et la version 2.0. Si la configuration de votre serveur principal est très personnalisée, vous trouverez peut-être plus simple de modifier votre configuration existante pour l'adapter au Serveur HTTP Apache 2.0. Les utilisateurs ayant une configuration peu personnalisée devront transférer leurs changements vers la configuration par défaut 2.0.

10.2.2.1. Mappage du fichier UserDir

La directive `UserDir` est utilisée pour permettre à des URL telles que `http://example.com/~bob/` de se mapper dans un sous-répertoire, dans le répertoire courant de l'utilisateur `bob`, comme par exemple `/home/bob/public_html`. Cette particularité permettant à un éventuel agresseur de déterminer si un nom d'utilisateur donné est présent sur le système, la configuration par défaut pour le Serveur HTTP Apache 2.0 désactive cette directive.

Pour activer le mappage de `UserDir`, changez la directive dans le fichier `httpd.conf` de:

```
UserDir disable
```

en ce qui suit:

```
UserDir public_html
```

Pour plus d'informations sur le sujet, reportez-vous à la documentation suivante sur le site Web d'Apache Software Foundation à l'adresse http://httpd.apache.org/docs-2.0/mod/mod_userdir.html#userdir.

10.2.2.2. Journalisation

Les directives de journalisation suivantes ont été supprimées:

- `AgentLog`
- `RefererLog`
- `RefererIgnore`

Cependant, les journaux Agent et Referrer sont encore disponibles en utilisant les directives `CustomLog` et `LogFormat`.

Pour plus d'informations sur le sujet, reportez-vous à la documentation suivante sur le site Web d'Apache Software Foundation:

- http://httpd.apache.org/docs-2.0/mod/mod_log_config.html#customlog
- http://httpd.apache.org/docs-2.0/mod/mod_log_config.html#logformat

10.2.2.3. Indexation des répertoires

La directive désapprouvée `FancyIndexing` est désormais supprimée. La même fonctionnalité est disponible par le biais de l'*option* `FancyIndexing` à l'intérieur de la directive `IndexOptions`.

La nouvelle option `VersionSort` à la directive `IndexOptions` engendre le classement des fichiers contenant des numéros de version dans un ordre plus naturel. Par exemple, `httpd-2.0.6.tar` apparaît avant `httpd-2.0.36.tar` dans une page d'index de répertoires.

Les paramètres par défaut pour les directives `ReadmeName` et `HeaderName` ont été transférés des fichiers `README` et `HEADER` vers les fichiers `README.html` et `HEADER.html`.

Pour plus d'informations sur le sujet, reportez-vous à la documentation suivante sur le site Web d'Apache Software Foundation:

- http://httpd.apache.org/docs-2.0/mod/mod_autoindex.html#indexoptions
- http://httpd.apache.org/docs-2.0/mod/mod_autoindex.html#readmename
- http://httpd.apache.org/docs-2.0/mod/mod_autoindex.html#headername

10.2.2.4. Négociation du contenu

La directive `CacheNegotiatedDocs` retient désormais les critères `on` ou `off`. Les cas existants de `CacheNegotiatedDocs` devront être remplacés par `CacheNegotiatedDocs on`.

Pour plus d'informations sur le sujet, reportez-vous à la documentation suivante sur le site Web d'Apache Software Foundation:

- http://httpd.apache.org/docs-2.0/mod/mod_negotiation.html#cachenegotiateddocs

10.2.2.5. Documents d'erreur

Afin de pouvoir utiliser un message codé en dur avec la directive `ErrorDocument`, le message devrait apparaître entre guillemets (["]), plutôt que d'être seulement précédé par des guillemets, comme c'était le Serveur HTTP Apache 1.3.

Pour transférer un paramètre `ErrorDocument` vers le Serveur HTTP Apache 2.0, utilisez la structure suivante:

```
ErrorDocument 404 "The document was not found"
```

Notez bien la présence des guillemets à la fin de l'exemple de directive `ErrorDocument` ci-dessus.

Pour plus d'informations sur le sujet, reportez-vous à la documentation suivante sur le site Web d'Apache Software Foundation:

- http://httpd.apache.org/docs-2.0/mod/core.html#errordocument

10.2.3. Configuration des hôtes virtuels

Le contenu de tous les répertoires `<VirtualHost>` devraient être migrés de la même manière que la section du serveur principal, comme cela est décrit dans la Section 10.2.2.


Important

> Notez que la configuration d'un hôte virtuel SSL/TLS a été supprimée du fichier de configuration du serveur principal et ajoutée dans le fichier `/etc/httpd/conf.d/ssl.conf`.

Pour plus d'informations sur le sujet, reportez-vous au chapitre intitulé *Configuration du serveur sécurisé HTTP Apache* du *Guide de personnalisation de Red Hat Linux* et à la documentation en ligne disponible à l'adresse:

- http://httpd.apache.org/docs-2.0/vhosts/

10.2.4. Modules et Serveur HTTP Apache 2.0

Dans la version 2.0 du Serveur HTTP Apache, le système des modules a été modifié afin de permettre aux modules d'être liés ensemble ou combinés de façons nouvelles et intéressantes. Les scripts *'Common Gateway Interface'* (*CGI*), par exemple, sont capables de générer des documents HTML analysés par le serveur qui peuvent ensuite être traités par `mod_include`. Grâce à ceci, il existe désormais un très grand nombre de possibilités quant à la façon dont les modules peuvent être combinés pour atteindre un objectif spécifique.

Cette opération est possible car chaque requête est servie par un seul module *'handler'*, suivi d'aucun ou de plusieurs modules *filter*.

Sous la version 1.3 du Serveur HTTP Apache, par exemple, un script PHP sera traité dans son intégralité par le module PHP. Sous la version 2.0 du Serveur HTTP Apache, en revanche, la requête est initialement *traitée* par le module mémoire ('core') — qui sert des fichiers statiques — et est ensuite *filtré* par le module PHP.

L'explication exacte de l'utilisation de cette fonction particulière et de toutes les autres nouvelles fonctions du Serveur HTTP Apache 2.0, va bien au-delà de la portée de ce document; toutefois, la conversion a des ramifications non-négligeables si vous avez utilisé la directive `PATH_INFO` pour un document traité par un module qui est désormais traité comme un filtre car chaque directive contient des informations de chemin non significatives après le vrai nom de fichier. Le module mémoire, qui traite initialement la requête, ne comprend pas par défaut `PATH_INFO` et renverra des erreurs de types `404 Not Found` pour les requêtes qui contiennent de telles information. Vous pouvez également utiliser la directive `AcceptPathInfo` pour obliger le module mémoire à accepter les requêtes contenant `PATH_INFO`.

Ci-dessous figure un exemple de cette directive:

```
AcceptPathInfo on
```

Pour plus d'informations sur le sujet, reportez-vous à la documentation suivante sur le site Web d'Apache Software Foundation:

- http://httpd.apache.org/docs-2.0/mod/core.html#acceptpathinfo
- http://httpd.apache.org/docs-2.0/handler.html
- http://httpd.apache.org/docs-2.0/filter.html

10.2.4.1. Module `mod_ssl`

La configuration de `mod_ssl` a été transférée du fichier `httpd.conf` au fichier `/etc/httpd/conf.d/ssl.conf`. Pour que ce dernier soit chargé et permette ainsi à `mod_ssl` de fonctionner correctement, la déclaration `Include conf.d/*.conf` doit figurer dans le fichier `httpd.conf`, comme le décrit la Section 10.2.1.3.

Dans les hôtes virtuels, les directives `ServerName` doivent explicitement spécifier le numéro de port.

Ci-dessous figure un exemple de la directive du Serveur HTTP Apache 1.3:

```
<VirtualHost _default_:443>
    # General setup for the virtual host
    ServerName ssl.example.name
    ...
</VirtualHost>
```

Pour transférer ce paramètre vers le Serveur HTTP Apache 2.0, utilisez la structure suivante:

```
<VirtualHost _default_:443>
    # General setup for the virtual host
    ServerName ssl.host.name:443
    ...
</VirtualHost>
```

Pour plus d'informations sur le sujet, reportez-vous à la documentation suivante sur le site Web d'Apache Software Foundation:

- http://httpd.apache.org/docs-2.0/mod/mod_ssl.html
- http://httpd.apache.org/docs-2.0/vhosts/

10.2.4.2. Module mod_proxy

Les instructions de contrôle d'accès proxy sont maintenant placées dans un bloc <Proxy> plutôt que dans un répertoire <Directory proxy:>.

La fonctionnalité de stockage temporaire de l'ancien mod_proxy a été divisée dans les trois modules suivants:

- mod_cache
- mod_disk_cache
- mod_file_cache

Ceux-ci utilisent généralement les mêmes directives ou des directives similaires aux versions plus anciennes du module mod_proxy.

Pour plus d'informations sur le sujet, reportez-vous à la documentation suivante sur le site Web d'Apache Software Foundation:

- http://httpd.apache.org/docs-2.0/mod/mod_proxy.html

10.2.4.3. Module mod_include

Le module mod_include fonctionne désormais comme un filtre et, pour cette raison, est activé d'une façon différente. Reportez-vous à la Section 10.2.4 pour obtenir de plus amples informations sur les filtres.

Ci-dessous figure un exemple de la directive du Serveur HTTP Apache 1.3:

```
AddType text/html .shtml
AddHandler server-parsed .shtml
```

Pour transférer ce paramètre vers le Serveur HTTP Apache 2.0, utilisez la structure suivante:

```
AddType text/html .shtml
AddOutputFilter INCLUDES .shtml
```

Notez que, comme auparavant, la directive Options +Includes est toujours nécessaire pour la section <Directory>, répertoire-conteneur, ou dans un fichier .htaccess.

Pour plus d'informations sur le sujet, reportez-vous à la documentation suivante sur le site Web d'Apache Software Foundation:

- http://httpd.apache.org/docs-2.0/mod/mod_include.html

10.2.4.4. Modules mod_auth_dbm et mod_auth_db

Le Serveur HTTP Apache 1.3 prenait en charge deux modules d'authentification, à savoir, mod_auth_db et mod_auth_dbm, qui utilisaient respectivement les bases de données Berkeley et DBM. Ces modules ont été rassemblés dans un seul module nommé mod_auth_dbm dans la version 2.0 du Serveur HTTP Apache, pouvant accéder à plusieurs formats de base de données différents. Pour migrer à partir du fichier mod_auth_db, fichiers de configuration devront être adaptés en remplaçant AuthDBUserFile et AuthDBGroupFile par les équivalents de mod_auth_dbm: AuthDBMUserFile et AuthDBMGroupFile. La directive AuthDBMType DB doit également être ajoutée pour préciser le type de de fichier de base de données utilisé.

Ci-dessous figure un exemple de la configuration mod_auth_db pour le Serveur HTTP Apache 1.3:

```
<Location /private/>
  AuthType Basic
  AuthName "My Private Files"
  AuthDBUserFile /var/www/authdb
  require valid-user
</Location>
```

Pour transférer ce paramètre vers la version 2.0 du Serveur HTTP Apache, utilisez la structure suivante:

```
<Location /private/>
  AuthType Basic
  AuthName "My Private Files"
  AuthDBMUserFile /var/www/authdb
  AuthDBMType DB
  require valid-user
</Location>
```

Notez que la directive `AuthDBMUserFile` peut également être utilisée dans des fichiers `.htaccess`.

Dans la version 2.0 du Serveur HTTP Apache, le script Perl `dbmmanage` utilisé pour manipuler des bases de données avec nom d'utilisateur et mot de passe, a été remplacé par `htdbm`. Le programme `htdbm` offre des fonctionnalités équivalentes et, tout comme le module `mod_auth_dbm` peut exploiter une grande variété de formats de bases de données; l'option `-T` peut être utilisée sur la ligne de commande pour spécifier le format à utiliser.

Tableau 10-1 montre comment migrer d'un format de base de données DBM vers `htdbm`, un format utilisant `dbmmanage`.

Action	Commande dbmmanage (1.3)	Équivalent de la commande htdbm (2.0)
Ajoute l'utilisateur à la base de données (en utilisant le mot de passe donné)	`dbmmanage authdb add username password`	`htdbm -b -TDB authdb username password`
Ajoute l'utilisateur à la base de données (invite à fournir le mot de passe)	`dbmmanage authdb adduser username`	`htdbm -TDB authdb username`
Retire l'utilisateur de la base de données	`dbmmanage authdb delete username`	`htdbm -x -TDB authdb username`
Répertorie les utilisateur s dans la base de données	`dbmmanage authdb view`	`htdbm -l -TDB authdb`
Vérifie un mot de passe	`dbmmanage authdb check username`	`htdbm -v -TDB authdb username`

Tableau 10-1. Migration de `dbmmanage` vers `htdbm`

Les options `-m` et `-s` fonctionnent avec `dbmmanage` et `htdbm`, permettant d'utiliser les algorithmes MD5 ou SHA1 respectivement, pour hacher des mots de passe.

Quand vous créez une nouvelle base de données avec `htdbm`, l'option `-c` doit être utilisée.

Pour plus d'informations sur le sujet, reportez-vous à la documentation suivante sur le site Web d'Apache Software Foundation:

- http://httpd.apache.org/docs-2.0/mod/mod_auth_dbm.html

10.2.4.5. Module **mod_perl** Module

La configuration du module mod_perl a été transférée du fichier httpd.conf vers le fichier /etc/httpd/conf.d/perl.conf. Pour que ce fichier soit chargé et permette ainsi à mod_perl de fonctionner correctement, la déclaration Include conf.d/*.conf doit figurer dans le fichier httpd.conf, comme le décrit la Section 10.2.1.3.

Les occurrences de Apache:: contenues dans votre fichier httpd.conf doivent être remplacées par ModPerl::. En outre, la façon dont les pilotes ('handlers') sont enregistrés a été modifiée.

Ci-dessous figure un exemple de la configuration du Serveur HTTP Apache 1.3 pour le module mod_perl:

```
<Directory /var/www/perl>
    SetHandler perl-script
    PerlHandler Apache::Registry
    Options +ExecCGI
</Directory>
```

Ci-après se trouve l'équivalent pour le module mod_perl sous la version 2.0 du Serveur HTTP Apache:

```
<Directory /var/www/perl>
    SetHandler perl-script
    PerlModule ModPerl::Registry
    PerlHandler ModPerl::Registry::handler
    Options +ExecCGI
</Directory>
```

La plupart des modules pour mod_perl 1.x devraient fonctionner sans modification avec mod_perl 2.x. Les modules XS doivent être recompilés et des modifications mineures de Makefile seront peut-être également nécessaires.

10.2.4.6. Module **mod_python**

La configuration du module mod_python a été transférée du fichier httpd.conf vers le fichier /etc/httpd/conf.d/python.conf. Pour que ce dernier soit chargé et permette ainsi à mod_python de fonctionner correctement, la déclaration Include conf.d/*.conf doit figurer dans le fichier httpd.conf comme le décrit la Section 10.2.1.3.

10.2.4.7. PHP

La configuration de PHP a été transférée du fichier httpd.conf vers le fichier /etc/httpd/conf.d/php.conf. Pour que celui-ci soit chargé, la déclaration Include conf.d/*.conf doit figurer dans le fichier httpd.conf, comme le décrit la Section 10.2.1.3.

PHP fonctionne désormais comme un filtre et doit, par conséquent, être activé d'une manière différente. Reportez-vous à la Section 10.2.4 pour plus d'informations sur les filtres.

Sous la version 1.3 du Serveur HTTP Apache, PHP était exécuté en utilisant les directives suivantes:

```
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps
```

Sous la version 2.0 du Serveur HTTP Apache, utilisez plutôt les directives suivantes:

```
<Files *.php>
    SetOutputFilter PHP
    SetInputFilter PHP
```

```
</Files>
```

Dans PHP 4.2.0 et les versions postérieures, l'ensemble des variables prédéfinies par défaut et ayant généralement une portée globale a changé. L'entrée individuelle et les variables serveur ne sont plus directement placées par défaut dans la portée globale. Ce changement risque d'interrompre les scripts. Vous devrez peut-être revenir à l'ancien comportement en réglant `register_globals` sur `On` dans le fichier `/etc/php.ini`.

Pour plus d'informations sur le sujet et pour obtenir des détails sur les changements au niveau de la portée globale, reportez-vous à l'adresse suivante:

- http://www.php.net/release_4_1_0.php

10.3. Après l'installation

Une fois l'installation du paquetage `httpd` terminée, la documentation sur le Serveur HTTP Apache est disponible en installant le paquetage `httpd-manual` et en vous rendant à l'adresse http://localhost/manual/ ou en parcourant la documentation du Serveur HTTP Apache disponible en ligne à l'adresse suivante: http://httpd.apache.org/docs-2.0/.

La documentation du Serveur HTTP Apache contient une liste exhaustive de toutes les options de configuration et leurs descriptions complètes. Pour plus de commodité, ce guide fournit de brèves descriptions des directives de configuration utilisées par le Serveur HTTP Apache 2.0.

La version du Serveur HTTP Apache faisant partie de Red Hat Linux offre la possibilité de définir des serveurs Web sécurisés au moyen du cryptage SSL offert par les paquetages `mod_ssl` et `openssl`. Lorsque vous examinez le fichier de configuration, sachez qu'il contient aussi bien un serveur Web non-sécurisé et un serveur Web sécurisé. Le serveur Web sécurisé fonctionne comme un hôte virtuel, qui est configuré dans le fichier `/etc/httpd/conf.d/ssl.conf`. Pour obtenir de plus amples informations sur les hôtes virtuels, reportez-vous à la Section 10.8. Pour vous informer sur la configuration d'un hôte virtuel sur un serveur sécurisé, reportez-vous à la Section 10.8.1. Pour vous informer sur l'installation d'un serveur sécurisé HTTP Apache, reportez-vous au chapitre intitulé *Configuration du serveur sécurisé HTTP Apache* du *Guide de personnalisation de Red Hat Linux*.



Remarque

Red Hat, Inc. ne contient pas les extensions FrontPage car la licence Microsoft™ interdit d'inclure ces extensions dans le produit d'un fournisseur tiers. Pour obtenir plus d'informations sur les extensions FrontPage et le Serveur HTTP Apache, visitez l'adresse: http://www.rtr.com/fpsupport/.

10.4. Démarrage et arrêt de `httpd`

Le RPM `httpd` installe le script `/etc/rc.d/init.d/httpd` qui est accessible en utilisant la commande `/sbin/service`.

Pour démarrer votre serveur, en tant que super-utilisateur entrez la commande suivante:

```
/sbin/service httpd start
```

Pour arrêter votre serveur, en tant que super-utilisateur entrez la commande suivante:

```
/sbin/service httpd stop
```

L'option `restart` est une façon rapide d'arrêter et de redémarrer le Serveur HTTP Apache.

Pour redémarrer le serveur, en tant que super-utilisateur, entrez:

```
/sbin/service httpd restart
```



Remarque

Si vous exécutez le Serveur HTTP Apache en tant que serveur sécurisé, il est nécessaire de saisir le mot de passe du serveur lors de toute utilisation des options `start` ou `restart`.

Après avoir modifié le fichier `httpd.conf`, toutefois, il n'est pas nécessaire d'arrêter et de redémarrer votre serveur. En revanche, utilisez l'option `reload`.

Afin de recharger le fichier de configuration, en tant que super-utilisateur, entrez la commande suivante:

```
/sbin/service httpd reload
```



Remarque

Si vous exécutez le Serveur HTTP Apache en tant que serveur sécurisé, vous *n'aurez pas* besoin de saisir votre mot de passe lors de utilisation de l'option `reload` (recharger).

Par défaut, le service `httpd` ne démarrera *ne démarrera pas* automatiquement au démarrage. Pour configurer le service `httpd` de manière à ce qu'il se lance au démarrage, utilisez un utilitaire de script d'initialisation ('initscript'), comme `/sbin/chkconfig`, **/sbin/ntsysv** ou le programme **Outil de configuration des services**. Reportez-vous au chapitre intitulé *Contrôle d'accès aux services* du *Guide de personnalisation de Red Hat Linux* pour obtenir plus d'informations sur ces outils.



Remarque

Si vous exécutez le Serveur HTTP Apache en tant que serveur sécurisé, il faudra saisir le mot de passe de ce dernier après le démarrage de l'ordinateur, à moins que vous n'ayez créé un type spécifique de fichier de clés pour le serveur.

Pour vous informer sur l'installation d'un serveur sécurisé HTTP Apache, reportez-vous au chapitre intitulé *Configuration du serveur sécurisé HTTP Apache* du *Guide de personnalisation de Red Hat Linux*.

10.5. Directives de configuration dans `httpd.conf`

Le fichier de configuration du Serveur HTTP Apache est `/etc/httpd/conf/httpd.conf`. Le fichier `httpd.conf` est bien commenté et parle de lui-même. Sa configuration par défaut fonctionne dans la plupart des situations; cependant, il est important de vous familiariser avec certaines des options de configuration les plus importantes.


Avertissement

Avec la sortie du Serveur HTTP Apache 2.0, de nombreuses options de configuration ont changé. Si vous devez migrer un fichier de configuration version 1.3 vers le nouveau format, reportez-vous à la Section 10.2.

10.5.1. Astuces de configuration générales

Si vous devez configurer le Serveur HTTP Apache, modifiez `/etc/httpd/conf/httpd.conf` puis rechargez, redémarrez ou arrêtez le processus `httpd` comme l'explique la Section 10.4.

Avant de modifier `httpd.conf`, faites d'abord une copie de sauvegarde du fichier original. Ainsi, si vous commettez ensuite une erreur lors de la modification du fichier de configuration, vous pourrez utiliser la copie de sauvegarde pour résoudre les problèmes.

Si vous commettez une erreur et que votre serveur Web ne fonctionne pas correctement, vérifiez tout d'abord les modifications apportées au fichier `httpd.conf` afin de corriger toute faute de frappe.

Consultez ensuite le journal des erreurs du serveur Web, `/var/log/httpd/error_log`. Le journal des erreurs peut être quelque peu difficile selon votre expérience. Toutefois, si vous venez de rencontrer un problème, les dernières entrées du journal des erreurs devraient fournir certaines indications sur ce qui s'est produit.

Les sections suivantes contiennent de brèves descriptions des directives contenues dans le fichier `httpd.conf`. Ces descriptions ne sont pas exhaustives. Pour plus d'informations, reportez-vous à la documentation d'Apache fournie au format HTML à l'adresse http://localhost/manual/ ou en ligne à l'adresse suivante: http://httpd.apache.org/docs-2.0/.

Pour plus d'informations sur les directives `mod_ssl`, reportez-vous à la documentation fournie au format HTML à l'adresse http://localhost/mod/mod_ssl.html ou en ligne à l'adresse suivante: http://httpd.apache.org/docs-2.0/mod/mod_ssl.html.

10.5.2. `ServerRoot`

Le répertoire `ServerRoot` est le répertoire de niveau supérieur contenant les fichiers du serveur. Tant le serveur sécurisé que le serveur non-sécurisé établissent la directive `ServerRoot` à `"/etc/httpd"`.

10.5.3. `ScoreBoardFile`

`ScoreBoardFile` stocke les informations internes au processus serveur utilisées pour la communication entre le processus serveur parent et ses processus enfants. Red Hat Linux utilise la mémoire partagée pour stocker `ScoreBoardFile`, la valeur par défaut `/etc/httpd/logs/apache_runtime_status` n'est utilisée qu'en cas de secours.

10.5.4. `PidFile`

`PidFile` est le nom du fichier dans lequel le serveur consigne son identifiant de processus (PID). Le PID par défaut est `/var/run/httpd.pid`.

10.5.5. `Timeout`

`Timeout` définit, en secondes, la durée pendant laquelle le serveur attend des réceptions et des émissions en cours de communication. Plus spécifiquement, `Timeout` définit la durée pendant laquelle le serveur attend de recevoir une requête GET, la durée pendant laquelle il attend de recevoir des paquets TCP sur une requête POST ou PUT et la durée pendant laquelle il attend entre des ACK répondant aux paquets TCP. La valeur de `Timeout` est réglée à `300` secondes par défaut, ce qui est approprié dans la plupart des cas.

10.5.6. `KeepAlive`

`KeepAlive` définit si votre serveur autorisera plus d'un requête par connexion; cette directive peut servir à empêcher un client particulier d'utiliser une trop grande quantité des ressources du serveur.

Par défaut, la valeur de `Keepalive` est réglée sur `off`. Si la valeur de `Keepalive` est `on` et que le serveur devient très occupé, le serveur peut générer rapidement un maximum de processus enfants. Dans ce cas, les serveur sera considérablement ralenti. Si la directive `Keepalive` est activée, il est recommandé de donner à `KeepAliveTimeout` une valeur basse (reportez-vous à la Section 10.5.8 pour obtenir de plus amples informations sur la directive `KeepAliveTimeout`) et de contrôler le fichier journal `/var/log/httpd/error_log` du serveur. Ce fichier indique si le serveur est à cours de processus enfants.

10.5.7. `MaxKeepAliveRequests`

Cette directive définit le nombre maximum de requêtes autorisées par connexion persistante. Le groupe Apache Project recommande l'utilisation d'un paramétrage élevé, ce qui améliorera les performances du serveur. Par défaut, la valeur de `MaxKeepAliveRequests` est paramétrée sur `100`, ce qui est approprié pour la plupart des situations.

10.5.8. `KeepAliveTimeout`

`KeepAliveTimeout` définit la durée en secondes pendant laquelle votre serveur attendra, après avoir servi une requête, avant d'interrompre la connexion. Une fois que le serveur reçoit une requête, c'est la directive `Timeout` qui s'applique à sa place. Par défaut, la valeur donnée à `KeepAliveTimeout` est 15 secondes.

10.5.9. `MinSpareServers` and `MaxSpareServers`

Le Serveur HTTP Apache s'adapte de façon dynamique à la charge reçue en maintenant un nombre de processus serveur de rechange approprié en fonction du trafic. Le serveur vérifie le nombre de serveurs attendant une requête et en supprime s'ils sont plus nombreux que `MaxSpareServers` ou en crée s'ils sont moins nombreux que `MinSpareServers`.

La valeur par défaut donnée à `MinSpareServers` est `5`; la valeur par défaut attribuée à `MaxSpareServers` `20`. Ces paramètres par défaut devraient convenir à presque toutes les situations. Ne donnez pas à `MinSpareServers` une valeur très élevée car un tel choix créera une charge de traitement importante sur le serveur, même si le trafic est faible.

10.5.10. `StartServers`

`StartServers` définit le nombre de processus serveur créés au démarrage. Étant donné que le serveur Web supprime et crée des processus serveur, de façon dynamique en fonction de la charge du trafic, il n'est pas nécessaire de modifier ce paramètre. Votre serveur Web est configuré de manière à lancer huit processus serveur au démarrage.

10.5.11. `MaxClients`

`MaxClients` fixe une limite au nombre total de processus serveur, ou de clients connectés simultanément, pouvant s'exécuter en même temps. L'objectif principal de cette directive est d'éviter qu'un Serveur HTTP Apache surchargé n'entraîne le plantage de votre système d'exploitation. Pour des serveurs très solicités, cette valeur devrait être élevée. La valeur par défaut du serveur est 150. Il n'est pas recommandé d'attribuer à `MaxClients` une valeur supérieure à `256`.

10.5.12. `MaxRequestsPerChild`

`MaxRequestsPerChild` définit le nombre total de demandes que chaque processus serveur enfant sert avant de disparaître. L'attribution d'une valeur à `MaxRequestsPerChild` est importante afin d'éviter des pertes de mémoire induites par des processus longs. La valeur par défaut pour `MaxRequestsPerChild` pour le serveur est `1000`.

10.5.13. `Listen`

The `Listen` identifie les ports sur lesquels votre serveur Web acceptera les demandes entrantes. Par défaut, le Serveur HTTP Apache est paramétré pour écouter sur le port 80 pour les communications Web non sécurisées et (dans `/etc/httpd/conf.d/ssl.conf` définissant tout serveur sécurisé) sur le port 443 pour les communications Web sécurisées.

Si le Serveur HTTP Apache est configuré pour écouter sur un port dont le numéro est inférieur à 1024, il doit être lancé en tant que super-utilisateur. En revanche, pour les ports dont le numéro est égal ou supérieur à 1024, `httpd` peut être lancé en tant que simple utilisateur.

La directive `Listen` peut également être utilisée pour spécifier des adresses IP particulières sur lesquelles le serveur acceptera des connexions.

10.5.14. `Include`

`Include` permet d'inclure d'autres fichiers de configuration au moment du lancement.

Le chemin d'accès de ces fichiers de configuration peut être absolu ou relatif au `ServerRoot`.



Important

Pour que le serveur utilise individuellement des modules paquetés, tels que `mod_ssl`, `mod_perl` et `php`, la directive suivante doit être intégrée dans la `Section 1: Global Environment` du `httpd.conf`:

```
Include conf.d/*.conf
```

10.5.15. `LoadModule`

`LoadModule` est utilisée pour charger des modules DSO (de l'anglais 'Dynamic Shared Object', objet partagé dynamique). Pour plus d'informations sur le support DSO du Serveur HTTP Apache, y compris la manière précise d'utiliser la directive `LoadModule`, reportez-vous à la Section 10.7. Notez que l'ordre de chargement des modules *n'est plus important* avec le Serveur HTTP Apache 2.0. Reportez-vous à la Section 10.2.1.3 pour plus d'informations sur le support DSO du Serveur HTTP Apache 2.0.

10.5.16. `ExtendedStatus`

La directive `ExtendedStatus` contrôle le type d'informations sur l'état des serveurs produit par Apache, lorsque le module de commande `server-status` est appelé; le informations fournies peuvent être sommaires (`off`) ou détaillées (`on`). Le module de commande `Server-status` est appelé à l'aide des balises `Location`. Pour plus d'informations sur l'appel de `server-status` reportez-vous à la Section 10.5.63.

10.5.17. `IfDefine`

Les balises `<IfDefine>` et `</IfDefine>` entourent des directives de configuration. Elles s'appliquent si le test indiqué dans la balise `<IfDefine>` est vrai. Les directives sont ignorées si le test est faux.

Le test dans les balises `<IfDefine>` est un nom de paramètre (comme par exemple, `HAVE_PERL`). Si le paramètre est défini (c'est-à-dire spécifié comme argument de la commande de démarrage du serveur), le test est vrai. Dans ce cas, lorsque le serveur Web est démarré, le test est vrai et les directives contenues dans les balises `IfDefine` sont appliquées.

Par défaut, les balises `<IfDefine HAVE_SSL>` entourent les balises d'hôtes virtuels pour votre serveur sécurisé. Les balises `<IfDefine HAVE_SSL>` entourent également les directives `LoadModule` et `AddModule` pour `ssl_module`.

10.5.18. `User`

La directive `User` définit le nom d'utilisateur du processus serveur et détermine les fichiers auxquels le serveur peut avoir accès. Tous les fichiers inaccessibles à cet utilisateur seront également inaccessibles aux clients se connectant au Serveur HTTP Apache.

La valeur par défaut donnée à `User` est `apache`.



Remarque

Pour des raisons de sécurité, le Serveur HTTP Apache refuse d'être exécuté en tant que super-utilisateur (ou root). user.

10.5.19. `Group`

Spécifie le nom de groupe des processus du Serveur HTTP Apache.

La valeur par défaut attribuée à `Group` est `apache`.

10.5.20. ServerAdmin

Donnez comme valeur à la directive `ServerAdmin` l'adresse électronique de l'administrateur du serveur Web. Cette adresse électronique apparaîtra dans les messages d'erreur sur les pages Web générées par le serveur afin que les utilisateurs puissent signaler un problème en envoyant un message électronique à l'administrateur du serveur.

La valeur par défaut donnée à `ServerAdmin` est `root@localhost`.

Généralement, la valeur donnée à `ServerAdmin` est `Webmaster@example.com`. Vous pouvez ensuite créer un alias pour `Webmaster` au nom de la personne responsable du serveur Web dans `/etc/aliases` et exécuter `/usr/bin/newaliases`.

10.5.21. ServerName

Utilisez `ServerName` pour définir un nom d'hôte et un numéro de port (en accord avec la directive `Listen`) pour le serveur. La directive `ServerName` ne doit pas forcément correspondre au nom d'hôte de l'ordinateur. Par exemple, le serveur Web pourrait être `www.example.com` bien que le nom d'hôte de l'ordinateur soit `foo.example.com`. La valeur spécifiée dans `ServerName` doit être un nom de domaine (ou DNS, de l'anglais 'Domain Name Service') valide qui peut être résolu par le système — ne vous contentez surtout pas d'en inventer un.

Ci-dessous figure un exemple de directive `ServerName`:

```
ServerName www.example.com:80
```

Lors de la détermination d'un `ServerName`, assurez-vous que son adresse IP et son nom de serveur sont bien inclus dans le fichier `/etc/hosts`.

10.5.22. UseCanonicalName

Lorsque la valeur attribuée à cette directive est `on`, elle configure le Serveur HTTP Apache de manière à ce qu'il se référence en utilisant les valeurs précisées dans les directives `ServerName` et `Port`. En revanche, lorsque la valeur de `UseCanonicalName` est `off`, le serveur emploiera la valeur utilisée le client envoyant la requête lorsqu'il fait référence à lui-même.

Par défaut, la valeur attribuée à `UseCanonicalName` est `off`.

10.5.23. DocumentRoot

`DocumentRoot` est le répertoire contenant la plupart des fichiers HTML qui seront servis en réponse aux requêtes. La valeur par défaut pour `DocumentRoot` aussi bien pour le serveur Web sécurisé que pour se serveur Web non-sécurisé est le répertoire `/var/www/html`. Par exemple, le serveur pourrait recevoir une demande pour le document suivant:

```
http://example.com/foo.html
```

Le serveur recherche le fichier suivant dans le répertoire par défaut:

```
/var/www/html/foo.html
```

Pour modifier `DocumentRoot` afin qu'il ne soit pas partagé par un serveur Web sécurisé et un serveur Web non-sécurisé, reportez-vous à la Section 10.8.

10.5.24. `Directory`

Les balises `<Directory /path/to/directory>` et `</Directory>` créent ce qu'on appelle un *conteneur* sont utilisées pour entourer un groupe de directives de configuration devant uniquement s'appliquer à ce répertoire et ses sous-répertoires. Toute directive applicable à un répertoire peut être utilisée à l'intérieur des balises `<Directory>`.

Par défaut, des paramètres très restrictifs sont appliqués au répertoire 'root' (`/`), à l'aide des directives `Options` (voir la Section 10.5.25) et `AllowOverride` (voir la Section 10.5.26). Dans cette configuration, tout répertoire du système ayant besoin de paramètres plus permissifs doit contenir explicitement ces paramètres.

Dans la configuration par défaut, un autre conteneur `Directory` est également configuré pour `DocumentRoot`; ce faisant, des paramètres moins rigides sont assignés à l'arbre de répertoire, de manière à ce que le Serveur HTTP Apache puisse avoir accès à des fichiers placés dans ce dernier.

Le conteneur `Directory` peut également être utilisé pour configurer des répertoires `cgi-bin` supplémentaires pour des applications côté-serveur en dehors du répertoire spécifié dans la directive `ScriptAlias` (reportez-vous à la Section 10.5.44 pour obtenir de plus amples informations sur la directive `ScriptAlias`).

Pour ce faire, le conteneur `Directory` doit déterminer l'option `ExecCGI` pour ce répertoire.

Par exemple, si les scripts CGI se trouvent dans `/home/my_cgi_directory`, ajoutez le conteneur `Directory` suivant au fichier `httpd.conf`:

```
<Directory /home/my_cgi_directory>
    Options +ExecCGI
</Directory>
```

Ensuite, la directive `AddHandler` doit être dé-commentée pour permettre l'identification des fichiers ayant une extension `.cgi` en tant que scripts CGI. Reportez-vous à la Section 10.5.59 pour obtenir des instructions sur le paramétrage de `AddHandler`.

Pour que cette opération se déroule parfaitement, il est nécessaire de donner la valeur 0755 aux permissions pour les scripts CGI et au chemin d'accès complet aux scripts.

10.5.25. `Options`

La directive `Options` contrôle les fonctions du serveur disponibles dans un répertoire particulier. Par exemple, en vertu des paramètres restrictifs spécifiés pour le répertoire root, `Options` est définie uniquement sur `FollowSymLinks`. Aucune fonction n'est activée, à l'exception du fait que le serveur est autorisé à suivre les liens symboliques dans le répertoire root.

Par défaut, dans le répertoire `DocumentRoot`, `Options` est paramétrée pour inclure `Indexes` et `FollowSymLinks`. `Indexes` permet au serveur de générer le contenu d'un répertoire si aucun `DirectoryIndex` (par exemple, `index.html`) n'est spécifié. `FollowSymLinks` permet au serveur de suivre des liens symboliques dans ce répertoire.



Remarque

Les déclarations `Options` de la section de configuration du serveur principal doit être copiée individuellement dans chaque conteneur `VirtualHost`. Reportez-vous à la Section 10.5.69 pour obtenir de plus amples informations sur les conteneurs `VirtualHost`.

10.5.26. AllowOverride

La directive `AllowOverride` définit si des `Options` peuvent être invalidées par les instructions d'un fichier `.htaccess`. Par défaut, tant le répertoire super-utilisateur que `DocumentRoot` sont réglés pour interdire les invalidations `.htaccess`.

10.5.27. Order

La directive `Order` contrôle simplement l'ordre dans lequel les directives `allow` et `deny` sont analysées. Le serveur est configuré pour analyser les directives `Allow` avant d'analyser les directives `Deny` pour votre répertoire `DocumentRoot`.

10.5.28. Allow

`Allow` spécifie le demandeur pouvant accéder à un répertoire donné. Le demandeur peut être `all`, un nom de domaine, une adresse IP, une adresse IP partielle, une paire réseau/masque réseau, etc. Le répertoire `DocumentRoot` est configuré pour permettre (`Allow`) les requêtes de quiconque (`all`), ainsi tout le monde peut y accéder.

10.5.29. Deny

`Deny` fonctionne selon le même principe que `Allow`, sauf que cette fois-ci, l'accès est refusé à un demandeur donné. Le `DocumentRoot` n'est pas configuré par défaut pour refuser (`Deny`) des requêtes provenant d'un demandeur quelconque.

10.5.30. UserDir

`UserDir` est le nom du sous-répertoire, au sein du répertoire personnel de chaque utilisateur, où devraient être placés les fichiers HTML personnels devant être servis par le serveur Web. Par défaut, la valeur attribuée à cette directive est `disable` (désactiver).

Dans le fichier de configuration par défaut, le nom du sous-répertoire est `public_html`. Par exemple, le serveur pourrait recevoir la requête suivante:

```
http://example.com/~nom-d'utilisateur/foo.html
```

Le serveur rechercherait le fichier:

```
/home/username/public_html/foo.html
```

Dans l'exemple ci-dessus, `/home/username/` est le répertoire personnel de l'utilisateur (notez que le chemin d'accès par défaut aux répertoires personnels des utilisateurs peut être différent sur votre système).

Assurez-vous que les autorisations relatives aux répertoires personnels des utilisateurs sont correctement définies. Les répertoires personnels des utilisateurs doivent être définis sur 0711. Les bits de lecture (r) et d'exécution (x) doivent être définis sur les répertoires `public_html` des utilisateurs (0755 fonctionnera). Les fichiers qui seront servis dans les répertoires `public_html` des utilisateurs doivent être définis sur au moins 0644.

10.5.31. DirectoryIndex

DirectoryIndex est la page servie par défaut lorsqu'un utilisateur demande un index de répertoire en insérant une barre oblique (/) à la fin d'un nom de répertoire.

Lorsqu'un utilisateur demande à accéder à la page http://*example*/*ce_répertoire*/, il reçoit soit la page DirectoryIndex si elle existe, soit une liste de répertoires générée par le serveur. La valeur par défaut pour DirectoryIndex est index.html et le type de mappe index.html.var. Le serveur essaie de trouver l'un de ces fichiers et renvoie le premier qu'il trouve. S'il ne trouve aucun de ces fichiers et que Options Indexes est paramétrée pour ce répertoire, le serveur génère et renvoie une liste, au format HTML, des fichiers et sous-répertoires contenus dans le répertoire (à moins que la fonctionnalité de listage des répertoires ne soit desactivée).

10.5.32. AccessFileName

AccessFileName nomme le fichier que le serveur doit utiliser pour les informations de contrôle d'accès dans chaque répertoire. La valeur par défaut est .htaccess.

Juste après la directive AccessFileName, une série de balises Files appliquent un contrôle d'accès à tout fichier commençant par .ht. Ces directives refusent l'accès Web à tous les fichiers .htaccess (ou d'autres commençant par .ht) pour des raisons de sécurité.

10.5.33. CacheNegotiatedDocs

Par défaut, votre serveur Web demande aux serveurs proxy de ne pas mettre en cache des documents négociés sur la base du contenu (c'est-à-dire qui peuvent changer avec le temps ou suite à l'entrée du demandeur). Si la valeur pour CacheNegotiatedDocs est paramétré sur on, cette fonction est désactivée et les serveurs proxy seront alors autorisés à mettre en cache des documents.

10.5.34. TypesConfig

TypesConfig nomme le fichier qui définit la liste par défaut des correspondances de type MIME (extensions de nom de fichier associées à des types de contenu). Le fichier TypesConfig par défaut est /etc/mime.types. Au lieu d'éditer /etc/mime.types, il est plutôt recommandé d'ajouter des types MIME à l'aide de la directive AddType.

Pour plus d'informations sur AddType, reportez-vous à la Section 10.5.58.

10.5.35. DefaultType

DefaultType définit un type de contenu par défaut pour le serveur Web à utiliser pour des documents dont les types MIME ne peuvent pas être déterminés. La valeur par défaut est text/plain.

10.5.36. IfModule

Les balises <IfModule> et </IfModule> créent un conteneur conditionnel dontles directives ne sont activés que si le module spécifié est chargé. Les directives placées entre les balises IfModule sont traitées dans l'un des deux cas suivants. Les directives sont traitées si le module contenu dans la balise de début <IfModule> est chargé. Ou, si un point d'exclamation ([!]) figure devant le nom du module, les directives ne sont traitées que si le module dans la balise <IfModule> *n'est pas* chargé.

Pour de plus amples informations sur les modules du Serveur HTTP Apache, reportez-vous à la Section 10.7.

10.5.37. HostnameLookups

HostnameLookups peut être paramétrée sur on, off ou double. Si HostnameLookups est paramétrée sur on, serveur résout automatiquement l'adresse IP pour chaque connexion. La résolution de l'adresse IP implique que le serveur établit une ou plusieurs connexions avec un DNS, rallongeant la durée des opérations. Si HostnameLookups être paramétrée sur double, le serveur établira une recherche DNS double inversée, rallongeant ainsi encore plus la durée des opérations.

Afin de conserver des ressources sur le serveur, la valeur par défaut pour HostnameLookups est off.

Si des noms d'hôtes sont nécessaires dans les fichiers journaux de serveur, songez à exécuter l'un des nombreux outils conçus pour analyser les fichiers journaux; ces derniers effectuent des recherches DNS non seulement de manière plus efficace mais également en masse lors de la rotation des fichiers journaux de serveur Web.

10.5.38. ErrorLog

ErrorLog spécifie le fichier dans lequel sont consignées les erreurs du serveur. La valeur par défaut pour cette directive est /var/log/httpd/error_log.

10.5.39. LogLevel

LogLevel définit le niveau de détail des messages d'erreur devant s'appliquer aux journaux des erreurs. Les valeurs possible de LogLevel sont (du niveau le moins détaillé au niveau le plus détaillé) emerg, alert, crit, error, warn, notice, info ou debug. La valeur par défaut pour LogLevel est warn.

10.5.40. LogFormat

La directive LogFormat détermine le format devant s'appliquer aux fichiers journaux des différents serveurs Web. Le LogFormat utilisé dépend en fait des paramètres attribués dans la directive CustomLog (voir la Section 10.5.41).

Ci-dessous figurent les options de format s'appliquant si la valeur de la directive CustomLog est combined:

%h (adresse IP de l'hôte distant ou nom d'hôte)

Répertorie l'adresse IP distante du client demandeur. Si la valeur de HostnameLookups est on, le nom d'hôte du client est enregistré à moins que le DNS ne puisse le fournir.

%l (rfc931)

Option non-utilisée. Un tiret ([-]) apparaît à sa place dans le fichier journal.

%u (utilisateur authentifié)

Si l'authentification devait être nécessaire, le nom d'utilisateur du demandeur serait enregistré. De manière générale cette option n'est pas utilisée et un tiret ([-]) figure à sa place dans le fichier journal.

%t (date)

Enregistre la date et l'heure de la requête.

%r (chaîne de demandes)

Enregistre la chaîne de demandes telle qu'elle est venue du navigateur ou du client.

`%s` (état)

Enregistre le code d'état HTTP renvoyé à l'hôte client.

`%b` (octets)

Enregistre la taille du document.

`%\"%{Referer}i\"` (referrer)

Enregistre l'URL de la page Web qui lie à la demande courante de l'hôte client.

`%\"%{User-Agent}i\"` (utilisateur-agent)

Enregistre le type de navigateur Web effectuant la requête.

10.5.41. CustomLog

`CustomLog` identifie le fichier journal et le format du fichier journal. Par défaut, l'enregistrement se fait dans le fichier `/var/log/httpd/access_log`.

Le format par défaut pour `CustomLog` est `combined`. L'extrait ci-dessous illustre le format de fichier journal `combined`:

```
remotehost rfc931 user date "request" status bytes referrer user-agent
```

10.5.42. ServerSignature

La directive `ServerSignature` ajoute une ligne contenant la version du Serveur HTTP Apache et le nom du serveur (`ServerName`) pour tout document créé par un serveur, comme par exemple, les messages d'erreur renvoyés aux clients. La valeur par défaut pour `ServerSignature` est `on`.

La valeur de cette directive peut être `off` ou `EMail`. La valeur `EMail`, ajoute une référence HTML `mailto:ServerAdmin` à la ligne de signature des réponses produites automatiquement par le système.

10.5.43. Alias

Le paramètre `Alias` permet d'accéder aux répertoires se trouvant en dehors du répertoires `DocumentRoot`. Toute URL se terminant par l'alias sera automatiquement convertie en chemin d'accès vers l'alias. Par défaut, un alias pour un répertoire `icons` est déjà configuré. Un répertoire `icons` est accessible par le serveur Web, mais le répertoire n'est pas dans `DocumentRoot`.

10.5.44. ScriptAlias

La directive `ScriptAlias` définit l'endroit où se trouvent les scripts CGI. D'une manière générale, il est préférable de ne pas laisser de scripts CGI dans `DocumentRoot`, où ils peuvent être consultés comme des documents en texte. C'est pour cette raison qu'il existe un répertoire spécial, en dehors du répertoire `DocumentRoot`, contenant des exécutables et scripts côté-serveur, désigné par la directive `ScriptAlias` directive. Cette dernière est connue sous le nom `cgi-bin` et prend `/var/www/cgi-bin/` comme valeur par défaut.

Il est possible de créer des répertoires pour stocker des exécutables en dehors du répertoire `cgi-bin`. Pour de plus amples information sur la manière de procéder, reportez-vous à la Section 10.5.59 et à la Section 10.5.24.

10.5.45. `Redirect`

Lorsqu'une page Web est déplacée, `Redirect` peut être utilisée pour mapper l'ancienne URL sur une autre URL. Le format est le suivant:

```
Redirect /<ancien-chemin>/<nom-de-fichier> http://<domaine-actuel>/<chemin-actuel>/<nom-
de-fichier>
```

Dans cet exemple, remplacez *<ancien-chemin>* par les informations de l'ancien-chemin vers *<nom-de-fichier>* et *<domaine-actuel>* et *<chemin-actuel>* par les informations du domaine et chemin actuels pour *<nom-de-fichier>*.

Dans cet exemple, toute requête pour *<nom-de-fichier>* à l'ancien emplacement est automatiquement redirigée vers le nouvel emplacement.

Pour obtenir des informations sur les techniques de redirection, utilisez le module `mod_rewrite` inclus dans le Serveur HTTP Apache. Pour de plus amples informations sur la configuration du module `mod_rewrite`, reportez-vous à la documentation d'Apache Software Foundation disponible en ligne à http://httpd.apache.org/docs-2.0/mod/mod_rewrite.html.

10.5.46. `IndexOptions`

`IndexOptions` contrôle l'apparence des listes de répertoire générées par le serveur, en ajoutant entre autres, des icônes et des descriptions de fichier. Si `Options Indexes` est définie (voir la Section 10.5.25), le serveur Web génère une liste des répertoires lorsqu'il reçoit une requête HTTP pour un répertoire sans index.

Le serveur Web recherche tout d'abord, dans le répertoire demandé un fichier correspondant aux noms énumérés dans la directive `DirectoryIndex` (généralement, `index.html`). Si le serveur Web ne trouve aucun fichier `index.html`, le Serveur HTTP Apache génère une liste HTML des répertoires correspondant au répertoire demandé. L'apparence de cette liste de répertoires est contrôlée, en partie, par la directive `IndexOptions`.

La valeur de la configuration par défaut est `FancyIndexing`. Ainsi, un utilisateur peut réorganiser une liste de répertoires en cliquant sur les en-têtes des colonnes. En cliquant deux fois sur la même en-tête, le classement passera d'ordre ascendant à un ordre descendant. La valeur `FancyIndexing` affiche également différentes icônes selon les types de fichiers, et ce en fonctions de leur extension.

Si l'option `AddDescription` est utilisée avec `FancyIndexing`, une brève description du fichier sera incluse dans les listes de répertoires générées par le serveur.

`IndexOptions` comprend un certain nombre d'autres paramètres qui peuvent être définis pour contrôler l'apparence des répertoires générés par le serveur. Les paramètres incluent `IconHeight` et `IconWidth`, pour faire en sorte que le serveur inclue des balises HTML `HEIGHT` et `WIDTH` pour les icônes dans les pages Web générées par le serveur; `IconsAreLinks`, pour faire en sorte que les icônes agissent comme une partie de l'ancre du lien HTML, en même temps que le nom de fichier, et autres.

10.5.47. `AddIconByEncoding`

Cette directive nomme des icônes qui s'affichent par fichier avec codage MIME, dans des listes de répertoires générées par le serveur. Par exemple, le serveur Web est paramétré par défaut pour afficher l'icône `compressed.gif` à côté des fichiers codés MIME x-compress et x-gzip dans des listes de répertoire générées par le serveur.

10.5.48. AddIconByType

Cette directive nomme des icônes qui s'affichent à côté des fichiers avec des types MIME dans des listes de répertoire générées par serveur. Par exemple, le serveur est paramétré pour afficher l'icône `text.gif` à côté de fichiers avec un type MIME `text`, dans des listes de répertoire générées par le serveur.

10.5.49. AddIcon

`AddIcon` spécifie l'icône à afficher dans les listes de répertoire générées par le serveur pour des fichiers avec certaines extensions. Par exemple, le serveur Web est paramétré pour afficher l'icône `binary.gif` pour les fichiers portant les extensions `.bin` ou `.exe`.

10.5.50. DefaultIcon

`DefaultIcon` spécifie l'icône à afficher dans les listes de répertoire générées par le serveur pour les fichiers pour lesquels aucune autre icône n'est spécifiée. Le fichier image `unknown.gif` est la valeur par défaut.

10.5.51. AddDescription

Lors de l'utilisation de `FancyIndexing` comme paramètre de `IndexOptions`, la directive `AddDescription` peut être utilisée pour afficher des descriptions spécifiées par l'utilisateur pour certains fichiers ou pour certains types de fichiers dans des listes de répertoire générées par le serveur. La directive `AddDescription` prend en charge les fichiers de listes spécifiques, les expressions à caractères génériques ('wildcards') ou les extensions de fichiers.

10.5.52. ReadmeName

`ReadmeName` nomme le fichier qui, s'il existe dans le répertoire, est ajouté à la fin des listes de répertoire générées par serveur. Le serveur Web commence par essayer d'inclure le fichier comme document HTML, puis essaie de l'inclure comme simple texte. Par défaut, `ReadmeName` est paramétré sur `README.html`.

10.5.53. HeaderName

`HeaderName` nomme le fichier qui, s'il existe dans le répertoire, est ajouté au début des listes de répertoire générées par serveur. Comme `ReadmeName`, le serveur essaie, si possible, de l'inclure sous la forme d'un document HTML ou, sinon, comme simple texte.

10.5.54. IndexIgnore

`IndexIgnore` affiche une liste d'extensions de fichier, de noms de fichier partiels, d'expressions contenant des caractères génériques ('wildcards') ou de noms de fichiers complets. Le serveur Web n'inclura dans les listes de répertoire générées par serveur, aucun fichier correspondant à l'un de ces paramètres.

10.5.55. AddEncoding

`AddEncoding` nomme des extensions de nom de fichier qui devraient spécifier un type de codage particulier. Il est également possible d'utiliser `AddEncoding` pour donner l'instruction à certains navigateurs de décompresser certains fichiers lors de leur téléchargement.

10.5.56. AddLanguage

`AddLanguage` associe des extensions de nom de fichiers à des langues spécifiques. Cette directive est très utilisée pour le Serveur HTTP Apache (ou plusieurs) qui sert des contenus dans une multitude de langues et ce, en fonction de la préférence linguistique définie sur le navigateur client.

10.5.57. LanguagePriority

`LanguagePriority` permet de déterminer l'ordre de préférence des langues, au cas aucune préférence linguistique ne serait paramétrée sur le navigateur client.

10.5.58. AddType

Utilisez la directive `AddType` pour définir des paires de type MIME et d'extension de fichier. Par exemple, avec PHP4, utilisez la directive `AddType` pour permettre au serveur Web de reconnaître les fichiers portant des extensions PHP (`.php4`, `.php3` `.phtml` `.php`) comme des types MIME PHP. La directive ci-dessous indique au Serveur HTTP Apache de reconnaître l'extension de fichier `.shtml`:

```
AddType text/html .shtml
AddHandler server-parsed .shtml
```

10.5.59. AddHandler

`AddHandler` mappe des extensions de fichier sur des modules de commande spécifiques. Par exemple, le module de commande `cgi-script` peut être utilisé en association avec l'extension `.cgi` pour traiter automatiquement un fichier dont le nom se termine par `.cgi` comme un script CGI. L'exemple suivant est un exemple de directive `AddHandler` pour l'extension `.cgi`.

```
AddHandler cgi-script .cgi
```

Cette directive actives les scripts CGI en dehors du répertoire `cgi-bin` afin qu'ils puissent fonctionner dans tout répertoire se trouvant sur le serveur, ayant l'option `ExecCGI` au sein du conteneur de répertoires. Reportez-vous à la Section 10.5.24 pour obtenir plus d'informations sur la définition de l'option `ExecCGI` pour un répertoire.

Outre son utilisation avec les scripts CGI, la directive `AddHandler` sert aussi au traitement de fichiers HTLM et imagemap analysés par le serveur.

10.5.60. Action

`Action` spécifie l'association d'un type MIME à un CGI, de sorte que toute requête d'un fichier de ce type déclenche l'exécution d'un script CGI particulier.

10.5.61. `ErrorDocument`

La directive `ErrorDocument` associe un code de réponse HTTP à un message ou à une URL qui sera renvoyé au client. Par défaut, le serveur Web renvoie un simple message d'erreur (habituellement obscur), lorsqu'une d'erreur se produit. Au lieu de ce paramétrage par défaut, il est possible d'utiliser la directive `ErrorDocument` pour forcer le serveur Web à renvoyer à la place, un message personnalisé ou à rediriger le client vers une URL locale ou externe.



Important

Pour que le message soit valide, il *doit* se trouver entre guillemets (["]).

10.5.62. `BrowserMatch`

La directive `BrowserMatch` permet au serveur de définir des variables d'environnement ou de prendre des mesures appropriées en fonction du champ d'en-tête Utilisateur-Agent HTTP — qui identifie le type de navigateur du client. Par défaut, le serveur Web utilise `BrowserMatch` pour refuser des connexions à certains navigateurs présentant des problèmes connus de même que pour désactiver les keepalives et vidages d'en-tête HTTP pour les navigateurs ayant des problèmes avec ces actions.

10.5.63. `Location`

Les balises `<Location>` et `</Location>` permettent de créer un conteneur dans lequel un contrôle d'accès basé sur l'URL peut être spécifié.

Par exemple, pour permettre aux personnes se connectant depuis le domaine du serveur de consulter des rapports sur l'état du serveur, utilisez les directives suivantes:

```
<Location /server-status>
    SetHandler server-status
    Order deny,allow Deny from all
    Allow from <.example.com>
</Location>
```

Remplacez `<.example.com>` par le nom de domaine de second niveau du serveur Web.

Pour fournir des rapports de configuration de serveur (y compris des modules installés et des directives de configuration) en réponse à des requêtes en provenance de votre domaine, utilisez les directives suivantes:

```
<Location /server-info>
    SetHandler server-info
    Order deny,allow
    Deny from all
    Allow from <.example.com>
</Location>
```

Ici encore, remplacez `<.example.com>` par le nom de domaine de second niveau du serveur Web.

10.5.64. ProxyRequests

Pour configurer le Serveur HTTP Apache de manière à ce qu'il fonctionne comme un serveur Proxy, supprimez le symbole dièse placé au début de la ligne <IfModule mod_proxy.c> pour charger le module mod_proxy et paramétrez la directive ProxyRequests sur On.

10.5.65. Proxy

Les balises <Proxy *> et </Proxy> permettent de créer un conteneur qui renferme un groupe de directives de configuration devant s'appliquer seulement au serveur proxy. À l'intérieur des balises <Proxy>, il est possible d'utiliser de nombreuses directives s'appliquant à un répertoire.

10.5.66. ProxyVia

La commande ProxyVia contrôle si une ligne d'en-tête HTTP Via: est envoyée en même temps que les demandes ou les réponses transitant par le serveur proxy Apache. L'en-tête Via: indique le nom d'hôte si ProxyVia a pour valeur On; spécifie le nom d'hôte et la version du Serveur HTTP Apache si la valeur retenue est Full; transfère toutes les lignes Via: inchangées si la valeur est Off et supprime les lignes Via: si la valeur est Block.

10.5.67. Directives cache

Un certain nombre de directives cache commentées sont fournies dans le fichier de configuration par défaut du Serveur HTTP Apache. Dans la plupart des situations, il suffit de supprimer le commentaire en retirant le symbole dièse ([#]) placé au début de la ligne. Toutefois, ci-après figure une liste de certaines des directives associées au cache ayant une grande importance:

- CacheRoot — définit le nom du répertoire qui contiendra les fichiers mis en cache. La valeur par défaut pour CacheRoot est le répertoire /var/httpd/proxy/.
- CacheSize — définit la quantité d'espace en kilo-octets (Ko) que le cache peut utiliser.La valeur par défaut pour CacheSize est 5 Ko.
- CacheGcInterval — définit la durée en heures devant s'écouler avant que les fichiers mis en cache ne soient supprimées. La valeur par défaut pour CacheGcInterval est 4 heures.
- CacheMaxExpire — définit la durée pendant laquelle les documents HTML mis en cache seront conservés (sans rechargement à partir du serveur Web dont ils proviennent). La valeur par défaut est de 24 heures.
- CacheLastModifiedFactor — paramètre la création d'une date d'expiration pour un document qui a été reçu du serveur d'origine sans date d'expiration définie. La valeur par défaut pour CacheLastModifiedFactor est établie à 0.1, ce qui signifie que la date d'expiration de tout document de ce type est égale à un dixième de la durée écoulée depuis la dernière modification du document.
- CacheDefaultExpire — détermine la durée en heures, de l'expiration d'un document qui a été reçu à l'aide d'un protocole ne prenant pas en charge les délais d'expiration. La valeur par défaut est établie à 1 heure.
- NoCache — établit une liste d'hôtes dont le contenu n'est pas mis en cache.

10.5.68. `NameVirtualHost`

La directive `NameVirtualHost` associe une adresse IP à un numéro de port, si nécessaire, pour tout hôte virtuel portant un nom. La configuration d'hôtes virtuels nommés permet à un Serveur HTTP Apache de servir différents domaines sans devoir pour ce faire utiliser de multiples adresses IP.



Remarque

L'utilisation de tout hôte virtuel nommé fonctionne *seulement* avec des connexions HTTP non-sécurisées. Si vous devez employer des hôtes virtuels avec un serveur sécurisé, utilisez plutôt des hôtes virtuels basés sur l'adresse IP.

Afin d'activer des hôtes virtuels basés sur le nom, supprimez le caractère de commentaire de la directive de configuration `NameVirtualHost` et ajoutez l'adresse IP correcte. Ajoutez ensuite des conteneurs `VirtualHost` supplémentaires pour chaque hôte virtuel.

10.5.69. `VirtualHost`

Des balises `<VirtualHost>` et `</VirtualHost>` permettent de créer un conteneur soulignant les caractéristiques d'un hôte virtuel. Le conteneur `<VirtualHost>` accepte la plupart des directives de configuration.

Un ensemble de conteneurs `VirtualHost` commentés est fourni dans `httpd.conf` et illustre l'ensemble minimum de directives de configuration nécessaire pour chaque hôte virtuel. Reportez-vous à la Section 10.8 pour obtenir de plus amples informations sur les hôtes virtuels.



Remarque

Tous les contextes des hôtes virtuels SSL ont été transférés dans le fichier `/etc/httpd/conf.d/ssl.conf`.

10.5.70. Directives de configuration SSL

Les directives SSL figurant dans le fichier `/etc/httpd/conf.d/ssl.conf` de votre serveur sont incluses pour permettre des communications Web sécurisées à l'aide de SSL et TLS.

10.5.70.1. `SetEnvIf`

La directive `SetEnvIf` permet de régler des variables d'environnement en fonction des en-têtes des connexions sécurisées entrantes. dans les demandes. Dans le fichier `/etc/httpd/conf.d/ssl.conf` fourni, elle sert à désactiver la fonction keep-alive HTTP et à autoriser SSL à fermer la connexion sans générer d'alerte de notification de fermeture de la part du navigateur client. Ce paramètre est nécessaire pour certains navigateurs qui n'interrompent pas la connexion SSL avec une grande fiabilité.

Pour obtenir de plus amples informations sur les directives SSL, à l'aide d'un navigateur, rendez-vous à l'une des adresses suivantes:

- http://localhost/manual/mod/mod_ssl.html

- http://httpd.apache.org/docs-2.0/mod/mod_ssl.html

Pour vous informer sur l'installation d'un serveur sécurisé HTTP Apache, reportez-vous au chapitre intitulé *Configuration du serveur sécurisé HTTP Apache* du *Guide de personnalisation de Red Hat Linux*.



Remarque

Les directives SSL, comme elles sont installées, sont configurées de manière appropriée pour la plupart des situations. Faîtes très attention lors de la modification des directives du serveur HTTP Apache car une mauvaise configuration peut être à l'origine de brèches de sécurité, rendant votre système vulnérable.

10.6. Modules par défaut

Serveur HTTP Apache est distribué avec un certain nombre de modules. Par défaut, les modules suivants sont installés et activés avec le paquetage `httpd` sur Red Hat Linux:

```
mod_access
mod_auth
mod_auth_anon
mod_auth_dbm
mod_auth_digest
mod_include
mod_log_config
mod_env
mod_mime_magic
mod_cern_meta
mod_expires
mod_headers
mod_usertrack
mod_unique_id
mod_setenvif
mod_mime
mod_dav
mod_status
mod_autoindex
mod_asis
mod_info
mod_cgi
mod_dav_fs
mod_vhost_alias
mod_negotiation
mod_dir
mod_imap
mod_actions
mod_speling
mod_userdir
mod_alias
mod_rewrite
mod_proxy
mod_proxy_ftp
mod_proxy_http
mod_proxy_connect
```

En outre, les modules suivants sont disponibles en installant des paquetages complémentaires:

```
mod_auth_mysql
mod_auth_pgsql
mod_perl
mod_python
mod_ssl
php
squirrelmail
```

10.7. Ajout de modules

Le Serveur HTTP Apache prend en charge les objets partagés dynamiques (ou *DSO* de l'anglais '*Dynamically Shared Objects*') ou des modules, qui peuvent être chargés facilement lors de l'exploitation selon les besoins.

L'Apache Project fournit en ligne une documentation complète sur les objets partagés dynamiques (DSO) à l'adresse suivante: http://httpd.apache.org/docs-2.0/dso.html. Sinon, si le paquetage `http-manual` est installée, de la documentation sur DSO se trouve à http://localhost/manual/mod/.

Pour que le Serveur HTTP Apache puisse utiliser un DSO, ce dernier doit être spécifié dans une directive `LoadModule` du répertoire `/etc/httpd/conf/httpd.conf`; si le module est fourni par un paquetage séparé, la ligne doit apparaître dans le fichier de configuration du module dans le répertoire `/etc/httpd/conf.d/`. Reportez-vous à la Section 10.5.15 pour obtenir de plus amples informations sur la directive `LoadModule`.

Lors de l'ajout ou de la suppression des modules du fichier `http.conf`, le Serveur HTTP Apache doit être rechargé et relancé, comme l'explique la Section 10.4.

Lors de la création d'un nouveau module, installez tout d'abord le paquetage `httpd-devel` car il contient les fichiers à inclure, les fichiers d'en-tête ainsi que l'application *Apache eXtenSion* (`/usr/sbin/apxs`), qui utilise les fichiers à inclure et les fichiers d'en-tête pour compiler les DSO.

Après l'écriture d'un module, utilisez la commande `/usr/sbin/apxs` pour compiler les sources de votre module en dehors de l'arbre source Apache. Pour obtenir de plus amples informations sur l'utilisation de la commande `/usr/sbin/apxs`, reportez-vous à la documentation Apache fournie en ligne à l'adresse suivante: http://httpd.apache.org/docs-2.0/dso.html ou consultez la page de manuel relative à `apxs`.

Une fois le module compilé, placez-le dans le répertoire `/usr/lib/httpd/`. Ajoutez ensuite une ligne `LoadModule` dans le fichier `httpd.conf` en suivant la structure suivante:

```
LoadModule <nom-module> <chemin/au/module.so>
```

Dans l'exemple ci-dessus, remplacez <*nom-module*> par le mon du module et <*chemin/au/module.so*> par le chemin d'accès au DSO.

10.8. Hôtes virtuels

La fonction des hôtes virtuels intégrée du Serveur HTTP Apache permet au serveur de servir des informations différentes en fonction de l'adresses IP, du nom d'hôte ou du port faisant l'objet de la requête. Un guide complet sur l'utilisation des hôtes virtuels est disponible en ligne à l'adresse suivante: http://httpd.apache.org/docs-2.0/vhosts/.

10.8.1. Configuration d'hôtes virtuels

La meilleure façon de créer un hôte virtuel nommés consiste à utiliser le conteneur d'hôte virtuel fournit dans `httpd.conf` à titre d'exemple.

L'exemple d'hôte virtuel se présente de la manière suivante:

```
#NameVirtualHost *
#
#<VirtualHost  *>
#    ServerAdmin Webmaster@dummy-host.example.com
#    DocumentRoot /www/docs/dummy-host.example.com
#    ServerName dummy-host.example.com
#    ErrorLog logs/dummy-host.example.com-error_log
#    CustomLog logs/dummy-host.example.com-access_log common
#</VirtualHost>
```

Pour activer la fonction d'hôte virtuel nommé, dé-commentez la ligne `NameVirtualHost` en retirant le symbole dièse (`#`) et en le remplaçant par l'astérisque (`*`) avec l'adresse IP attribuée à l'ordinateur.

Configurez ensuite un hôte virtuel, en dé-commentant et personnalisant le conteneur `<VirtualHost>`.

Sur la ligne `<VirtualHost>`, remplacez l'astérisque (`*`) par l'adresse IP du serveur. Remplacez aussi `ServerName` par to a par le nom d'un DNS *valide* assigné à l'ordinateur et configurez les autres directives selon les besoins.

Étant donné que le conteneur `<VirtualHost>` accepte presque toutes les directives disponibles dans le cadre de la configuration du serveur principal, sa capacité à être personnalisé est très élevée.


Astuce

> Si vous configurez un hôte virtuel et souhaitez qu'il contrôle un port non-défini par défaut, ce dernier doit être ajouté à la directive `Listen` dans la partie des paramètres globaux du fichier `/etc/httpd/conf/http.conf`.

Afin de pouvoir activer l'hôte virtuel venant d'être créé, le Serveur HTTP Apache doit être rechargé ou redémarré. Reportez-vous à la Section 10.4 pour obtenir des instructions sur ces opérations.

Des informations complètes sur la création et la configuration d'hôtes virtuels sur la base du nom ou de l'adresse IP sont fournies en ligne à l'adresse suivante: http://httpd.apache.org/docs-2.0/vhosts/.

10.8.2. Hôte virtuel du serveur Web sécurisé

Par défaut, le Serveur HTTP Apache est configuré aussi bien comme un serveur Web non-sécurisé que comme un serveur sécurisé. Les deux serveurs (non-sécurisé et sécurisé) utilisent la même adresse IP et le même nom d'hôte, mais contrôlent des ports différents, à savoir, 80 et 443 respectivement. Ce faisant, des communications aussi bien non-sécurisées que sécurisées peuvent être établies simultanément.

Il est important de savoir que les transmissions HTTP améliorées grâce à SSL monopolisent cependant plus de ressources que le protocole HTTP standard et que par conséquent, un serveur sécurisé sert moins de pages par seconde. Dans de telles conditions, il est recommandé de minimiser les informations disponibles à partir du serveur sécurisé, tout particulièrement sur un site Web très solicité.


Important

N'utilisez pas d'hôtes virtuels nommés de concert avec un serveur Web sécurisé car le protocole de transfert SSL intervient avant que la requête HTTP n'identifie l'hôte virtuel nommé approprié. Les hôtes virtuels nommés ne fonctionnent qu'avec un serveur Web non-sécurisé.

Les directives de configuration pour du serveur sécurisé se trouvent entre des balises d'hôte virtuel dans le fichier `/etc/httpd/conf.d/ssl.conf`.

Par défaut, les deux serveurs Web, sécurisé et non-sécurisé, partagent le même `DocumentRoot`. Il est cependant recommandé que `DocumentRoot` soit différent pour le serveur Web sécurisé.

Afin que le serveur Web non-sécurisé n'accepte plus de connexions, commentez la ligne qui se trouve dans `httpd.conf` et stipule `Listen 80` en ajoutant un symbole dièse au début de la ligne. Une fois cette opération terminée, le ligne ressemblera à l'extrait ci-dessous:

```
#Listen 80
```

Pour plus d'informations sur la configuration d'un serveur Web utilisant SSL, reportez-vous au chapitre intitulé *Configuration du serveur HTTP Apache sécurisé* du *Guide de personnalisation de Red Hat Linux*. Pour obtenir des astuces de configuration avancées, consultez la documentation d'Apache Software Foundation disponible en ligne aux adresses suivantes:

- http://httpd.apache.org/docs-2.0/ssl/.
- http://httpd.apache.org/docs-2.0/vhosts/

10.9. Ressources supplémentaires

Pour en savoir plus sur Serveur HTTP Apache, veuillez vous reporter aux ressources qui suivent.

10.9.1. Sites Web utiles

- http://httpd.apache.org — Le site Web officiel du Serveur HTTP Apache contenant de la documentation non seulement sur toutes les directives mais également sur tous les modules par défaut.
- http://www.modssl.org — Le cite Web officiel de `mod_ssl`.
- http://www.apacheweek.com — Une excellente ressource publiant en ligne toutes les semaines, des documents de toutes sortes en relation avec Apache.

10.9.2. Livres sur le sujet

- *Apache Desktop Reference* de Ralf S. Engelschall; Addison Wesley — Écrit par Ralf Engelschall, un membre d'Apache Software Foundation (ASF) et auteur de `mod_ssl`, *Apache Desktop Reference* est un guide de référence concis et exhaustif pour l'utilisation du Serveur HTTP Apache et plus spécialement pour sa compilation, sa configuration et son exécution. Ce livre est également disponible en ligne à http://www.apacheref.com/.
- *Professional Apache* de Peter Wainwright; Wrox Press Ltd — *Professional Apache* est un des nombreux livres de la collection "Programmer to Programmer" de la maison d'édition Wrox Press Ltd, destiné aussi bien aux administrateurs de serveurs Web aussi bien expérimentés que débutants.

- *Administering Apache* de Mark Allan Arnold; Osborne Media Group — Ce livre est destiné aux fournisseurs d'accès Internet désireux d'offrir des services plus sécurisés.
- *Apache Server Unleashed* de Richard Bowen, et al; SAMS BOOKS — Une source encyclopédique pour le Serveur HTTP Apache.
- *Apache Pocket Reference* d'Andrew Ford, Gigi Estabrook; O'Reilly — La dernière nouveauté de la collection O'Reilly Pocket Reference.



Chapitre 11. Courrier électronique

La naissance du courrier électronique (ou *email*) remonte au début des années 1960. La boîte à lettres était un fichier dans le répertoire personnel d'un utilisateur que seul ce dernier pouvait lire. Les application de courrier primitives ajoutaient des nouveaux messages de texte bas du fichier et l'utilisateur devait parcourir tout le fichier qui ne cessait de grandir afin de retrouver un message particulier. Ce système ne pouvait envoyer des messages qu'aux utilisateurs du système.

Le premier transfert réseau réel d'un courrier électronique eu lieu en 1971 lorsqu'un ingénieur informatique du nom de Ray Tomlinson envoya un message test entre deux ordinateurs via ARPANET — le précurseur de l'Internet. La communication par email devint rapidement très populaire, représentant 75 pour cent du trafic d'ARPANET en moins de deux ans.

Au fil du temps, les systèmes de courrier électronique basés sur des protocoles réseau standardisés ont évolués de telle manière qu'ils font partie de nos jours, des services les plus couramment utilisés sur l'Internet. Red Hat Linux offre de nombreuses applications avancées pour servir et accéder aux emails.

Ce chapitre examine d'une part les protocoles de courrier électronique utilisés à l'heure actuelle et d'autre part, certains des programmes de messagerie électroniques conçus pour envoyer et recevoir des emails.

11.1. Protocoles de courrier électronique

De nos jours, le courrier électronique délivré à l'aide d'une architecture client/serveur. Un message électronique est créé au moyen d'un programme client de messagerie électronique. Ce programme envoie ensuite le message à un serveur. Ce dernier transmet alors à son tour le message au serveur email du destinataire où il est fournit au client email du destinataire final.

Afin de rendre ce processus possible, une vaste gamme de protocoles réseau standard permettent à différents ordinateurs exécutant souvent différents systèmes d'exploitation et utilisant des programmes de messagerie électroniques différents, d'envoyer et de recevoir des emails.

Les protocoles suivants traités dans ce chapitre sont sont ceux le plus fréquemment utilisés pour le transfert de courrier électronique entre systèmes.

11.1.1. Protocoles de transfert de courrier électronique

La délivrance de courrier d'une application cliente au serveur et d'un serveur d'origine à un serveur de destination est traitée par le protocole nommé *'Simple Mail Transfer Protocol'* (ou *SMTP*) .

11.1.1.1. SMTP

L'objectif primaire de SMTP consiste à transférer le courrier électronique entre les serveurs de messagerie. Toutefois, il est également très important pour les clients de messagerie. Afin d'envoyer un email, le client envoie le message électronique à un serveur de messagerie sortant , qui à son tour contacte le serveur de messagerie de destination pour la délivrance du message. Dans de telles circonstances, il est nécessaire de spécifier un serveur SMTP lors de la configuraton d'un client email.

Sous Red Hat Linux, un utilisateur peut configurer un serveur SMTP sur l'ordinateur local afin qu'il traite la délivrance du courrier. Toutefois, il est également possible de configurer des serveurs SMTP distant pour le courrier sortant.

Il est important de noter ici que le protocole SMTP n'a pas besoin d'authentification pour fonctionner. Ainsi, toute personne utilisant l'Internet peut envoyer des emails à toute autre personne ou même à de

vastes groupes de personnes. C'est cette caractéristique de SMTP qui permet l'envoi de pourriel ('junk email') ou de *spam*. Les serveurs SMTP modernes essaient néanmoins de minimiser ce comportement en n'autorisant que les hôtes connus à accéder au serveur SMTP. Les serveurs n'imposant pas ce genre de restriction sont appelés serveurs *open relay*.

Red Hat Linux utilise Sendmail (`/usr/sbin/sendmail`) comme programme SMTP par défaut. Néanmoins, une application de serveur de messagerie plus simple appelée Postfix (`/usr/sbin/postfix`) est également disponible.

11.1.2. Protocoles d'accès au courrier

Pour obtenir le courrier électronique stocké sur les serveurs de messagerie, les applications client de messagerie utilisent deux protocoles primaires: *'Post Office Protocol'* (ou *POP*) et *'Internet Message Access Protocol'* (ou *IMAP*).

Contrairement à SMTP, ces deux protocoles exigent que les clients se connectant s'authentifie au moyen d'un nom d'utilisateur et d'un mot de passe. Par défaut, les mots de passe pour les deux protocoles sont transmis à travers le réseau de manière non-cryptée.

11.1.2.1. POP

Sous Red Hat Linux, `/usr/sbin/ipop3d` est le serveur POP par défaut et est inclus dans le paquetage `imap`. Lors de l'utilisation d'un serveur POP, le courrier électronique est téléchargé par des applications client de messagerie. Par défaut, la plupart des clients de messagerie POP sont configurés automatiquement pour supprimer les messages sur le serveur une fois le transfert effectué; néanmoins, cette configuration peut souvent être modifiée.

Le protocole POP est compatible à 100 % avec d'importantes normes de messagerie Internet, comme par exemple *'Multipurpose Internet Mail Extensions'* (ou *MIME*), qui permet l'envoi de fichiers joints.

Le protocole POP est le plus approprié pour les utilisateurs disposant d'un système sur lequel ils peuvent lire leurs courrier électronique. Il fonctionne également bien pour utilisateurs n'ayant pas une connexion continue à l'Internet ou au réseau sur lequel se trouve le serveur de messagerie. Malheureusement, les utilisateurs ayant des connexions réseau lentes, POP requiert que les programmes client, après authentification, téléchargent la totalité du contenu de chaque message. Cette opération peut être longue si certains messages contiennent des fichiers joints.

La version la plus courante du protocole POP standard est POP3.

Il existe néanmoins de nombreuses variantes moins utilisées du protocole POP:

- *APOP* — POP3 avec authentification MDS. Une portion codée du mot de passe de l'utilisateur est envoyée du client de messagerie au serveur plutôt que d'envoyer le mot de passe sous forme non-cryptée.
- *KPOP* — POP3 avec authentification Kerberos. Reportez-vous au Chapitre 17 pour obtenir plus d'informations sur l'authentification Kerberos.
- *RPOP* — POP3 avec authentification RPOP qui utilise un identificateur (ID) publié pour chaque utilisateur, semblable à un mot de passe, pour identifier les requêtes POP. Cependant, étant donné que cet ID n'est pas crypté, RPOP n'est pas plus sécurisé que le POP standard.

Pour une sécurité accrue, il est possible d'utiliser le cryptage *'Secure Socket Layer'* (*SSL*) pour l'authentification des clients et pour les sessions de transfert de données. Cette fonctionnalité peut être activée en utilisant le service `ipop3s` ou le programme `/usr/sbin/stunnel`. Reportez-vous à la Section 11.5.1 pour de plus amples informations.

11.1.2.2. IMAP

Sous Red Hat Linux, `/usr/sbin/imapd` est le serveur IMAP par défaut, fourni par le paquetage `imap`. Lors de l'utilisation d'un serveur de messagerie IMAP, le courrier électronique reste sur le serveur où les utilisateurs peuvent lire et supprimer les emails. IMAP permet également aux applications client de créer, renommer ou supprimer des répertoires de messagerie sur le serveur afin d'organiser ou de stocker le courrier électronique.

Le protocole IMAP est utile tout particulièrement pour les utilisateurs accédant à leur courrier électronique au moyen d'ordinateurs multiples. Ce protocole est également pratique pour les utilisateurs se connectant au serveur de messagerie par le biais d'une connexion lente, car seule l'information d'en-tête du message est téléchargée jusqu'à ce qu'il soit ouvert, économisant ainsi de la largeur de bande. En outre, l'utilisateur peut également supprimer des messages sans devoir les lire ou les télécharger.

Par commodité, les applications IMAP client peuvent mettre en cache localement des copies des messages afin que l'utilisateur puissent naviguer parmi des messages déjà lus même lorsqu'il n'est pas directement connecté au serveur IMAP.

IMAP, tout comme POP est compatible à 100 % avec d'importantes normes de messagerie Internet, telles que MIME (Multipurpose Internet Mail Extensions) pour permettre l'envoi de fichiers joints.

Pour une sécurité accrue, il est possible d'utiliser le cryptage *SSL* pour l'authentification des clients et pour les sessions de transfert de données. Cette fonctionnalité peut être activée en utilisant le service `imaps` ou le programme `/usr/sbin/stunnel` Reportez-vous à la Section 11.5.1 pour de plus amples informations.

D'autres clients et serveurs IMAP libres et commerciaux sont disponibles; un certain nombre d'entre eux développent encore plus les possibilités du protocole IMAP et fournissent des fonctionnalités supplémentaires. Une liste compréhensive de ces derniers est disponible en ligne à http://www.imap.org/products/longlist.htm.

11.2. Les différents types de programme de messagerie électronique

D'une manière générale, les applications de messagerie électronique se divisent en trois types et chacune d'elle peut appartenir à un ou plusieurs de ces type. Chaque type joue un rôle bien précis dans le processus de déplacement et de gestion des messages électroniques. Bien que la plupart des utilisateurs ne connaissent que le programme de courrier électronique qu'ils utilisent pour recevoir et envoyer des messages, chacun de ces trois types d'application est important pour assurer que les messages arrivent à la bonne destination.

11.2.1. Agent de transfert de courrier

L'*Agent de Transfert de Courrier* (*ATC*, ou MTA de l'anglais 'Mail Transfer Agent') sert à transférer des messages électroniques entre des hôtes utilisant SMTP. Un message peut requérir l'utilisation de plusieurs ATC lors de sa progression vers sa destination finale.

Alors que l'acheminement de messages entre ordinateurs peut sembler plutôt simple et direct, l'ensemble du processus permettant de décider si un ATC donné peut ou devrait accepter un message à envoyer est en fait assez complexe. De plus, en raison des problèmes créés par les spams, l'utilisation d'un ATC donné est généralement limitée par la configuration même de l'ATC ou par le manque d'accès au réseau de l'ATC.

Nombre d'ATC plus gros et plus complexes peuvent aussi être utilisés pour envoyer des messages. Toutefois, il ne faut pas confondre cette opération avec le vrai rôle d'un ATC. La seule raison pour laquelle les programmes client de messagerie peuvent envoyer des emails (comme ATC) réside dans le fait que l'hôte exécutant l'application ne dispose pas de son propre ATC. Cette situation s'applique tout particulièrement aux programmes client de messagerie faisant partie de systèmes d'exploitations

qui ne sont pas basés sur Unix. Cependant, ces programmes client de messagerie n'envoient que des messages de sortie à un ATC qu'ils sont autorisés à utiliser et n'acheminent pas directement le message au serveur de messagerie du destinataire.

Étant donné que Red Hat Linux installe deux ATC, à savoir Sendmail et Postfix, les programmes client de messagerie ne sont généralement pas sollicités pour agir en tant qu'ATC. Red Hat Linux inclut également un ACT avec un objectif bien spécifique, nommé Fetchmail.

Pour obtenir de plus amples informations sur Sendmail et Fetchmail, reportez-vous à la Section 11.3.

11.2.2. Agent de distribution du courrier (ADC)

Un *Agent de Distribution de Courrier* (*ADC* ou MDC de l'anglais 'Mail Delivery Agent') est utilisé par l'ATC pour distribuer le courrier arrivant dans la boîte à lettres de l'utilisateur approprié. Dans de nombreuses situations, l'ADC est en fait un *Agent de Distribution Locale* (*ADL*ou LDA de l'anglais 'Local Delivery Agent'), comme `mail` ou Procmail.

En fait, tout programme traitant un message pour la distribution jusqu'au point où il peut être lu par une application client de messagerie peut être considéré comme un ADC. Pour cette raison, certains ATC (comme Sendmail et Postfix) peuvent aussi jouer le rôle d'un ADC lorsqu'ils ajoutent de nouveaux messages électroniques au fichier spoule (ou 'spool') de courrier électronique d'un utilisateur local. En général, les ADC n'acheminent pas de messages entre les deux systèmes et ne fournissent pas une interface utilisateur; les ADC distribuent et classent les messages sur un ordinateur local pour qu'une application client de messagerie puissent y accéder.

11.2.3. Agent de gestion de courrier (AGC)

Un *Agent de Gestion de Courrier* (*AGC*, ou MUA de l'anglais 'Mail User Agent') est en fait une application client de messagerie. Un AGC est un programme qui, au minimum, permet à un utilisateur de lire et écrire des messages électroniques. De nombreux AGC peuvent récupérer des messages au moyen de protocoles POP ou IMAP, établissant des boîtes à lettres pour stocker les messages et envoyant des messages de sortie à un ATC.

Les AGC peuvent être graphiques, comme **Mozilla Mail**, ou peuvent avoir une simple interface à base de texte comme `mutt` ou `pine`.

11.3. Agent de transfert de courrier (ATC)

Red Hat Linux comprend deux Agent de transfert de courrier primaires, à savoir Sendmail et Postfix. Sendmail est configuré comme la valeur par défaut mais il est possible de remplacer facilement cette valeur par Postfix.



Astuce

Pour savoir comment passer d'un ATC par défaut Sendmail à un ATC par défaut Postfix, reportez-vous au chapitre intitulé *Configuration de l'Agent de transfert de courrier (ATC)* du *Guide de personnalisation de Red Hat Linux*.

Red Hat Linux inclut également un ATC doté d'une fonction particulière nommé Fetchmail; ce dernier est utilisé pour acheminer le courrier électronique d'un ACT distant à un ACT local.

Cette section examine de manière détaillé Sendmail et Fetchmail.

11.3.1. Sendmail

La tâche principale de Sendmail est de déplacer de façon sécurisée des messages électroniques entre des hôtes, utilisant généralement le protocole SMTP. Toutefois, Sendmail est hautement configurable, ce qui vous permet de contrôler presque tous les aspects du traitement des messages, y compris le protocole à utiliser. De nombreux administrateurs système choisissent d'utiliser Sendmail comme ATC en raison de sa puissance et de sa scalabilité.

11.3.1.1. Objectif et limites

Il est important de bien comprendre ce qu'est Sendmail et ce qu'il peut faire, de même que ce qu'il n'est pas. En cette période d'applications monolithiques jouant des rôles multiples, on pourrait penser que Sendmail est la seule application nécessaire pour exécuter un serveur de messagerie au sein d'une organisation. Techniquement parlant, ceci est vrai car Sendmail peut spouler du courrier sur vos répertoires utilisateur et accepter de nouveaux messages sortant pour les utilisateurs. Cependant, la plupart des utilisateurs désirent bien plus que le simple acheminement du courrier. Ils veulent en général interagir avec le courrier électronique à l'aide d'un AGC qui utilise POP ou IMAP pour télécharger leurs messages sur leur ordinateur local. Ou alors, ils pourraient préférer une interface Web pour avoir accès à leur boîte à lettres. Ces autres applications fonctionnent de concert avec Sendmail et SMTP, mais existent en réalité pour différentes raisons et peuvent fonctionner indépendemment les unes des autres.

L'explication de tout ce que Sendmail devrait et pourrait faire en fonction de sa configuration va bien au-delà de la portée de cette section. Étant donné le nombre d'options différentes et de réglages possibles, des volumes entiers ont été écrits pour expliquer toutes les possibilités de Sendmail et les façons de régler d'éventuels problèmes. Reportez-vous à la Section 11.6 pour obtenir une liste des ressources dédiées à Sendmail.

Cette section passe en revue les fichiers installés avec Sendmail par défaut et examine certaines modification de configuration élémentaires, y compris comment éviter de recevoir du pourriel (spam) et comment augmenter les capacités de Sendmail avec le protocole *'Lightweight Directory Access Protocol' (LDAP)*.

11.3.1.2. Installation de Sendmail par défaut

Le fichier exécutable de Sendmail est `/usr/sbin/sendmail`.

Le fichier de configuration de Sendmail, long et détaillé, est `/etc/mail/sendmail.cf`. Évitez d'éditer le fichier `sendmail.cf` directement. Pour apporter des modification à la configuration, éditez plutôt le fichier `/etc/mail/sendmail.mc`, sauvegardez le fichier original `/etc/mail/sendmail.cf` et utilisez ensuite le macroprocesseur `m4` qui est inclus pour créer un nouveau fichier `/etc/mail/sendmail.cf`. De plus ample informations sur la configuration de Sendmail sont disponibles dans la Section 11.3.1.3.

Divers fichiers de configuration Sendmail sont installés dans `/etc/mail/`, notamment:

- `access` — Spécifie les systèmes qui peuvent utiliser Sendmail pour le courrier électronique sortant.
- `domaintable` — Spécifie le mappage de noms de domaine.
- `local-host-names` — Spécifie les alias de l'hôte.
- `mailertable` — Spécifie des instructions qui écrasent le routage de domaines spécifiques.
- `virtusertable` — Spécifie une forme de dénomination par alias spécifique au domaine, ce qui permet à des domaines virtuels multiples d'être hébergés sur un ordinateur.

Plusieurs fichiers de configuration placés dans `/etc/mail/`, tels que `access`, `domaintable`, `mailertable` et `virtusertable`, doivent en fait stocker leurs informations dans des fichiers de base de

données avant que Sendmail puisse appliquer les modifications apportées à la configuration. Pour inclure les changements apportés à ces fichiers de configuration dans leurs fichiers de base de données, vous devez exécutez la commande:

```
makemap hash /etc/mail/<nom> < /etc/mail/<nom>
```

où <*nom*> doit être remplacé par le nom du fichier de configuration à convertir.

Par exemple, pour que tous les messages électroniques destinés au domaine `example.com` soit envoyés à <`bob@other-example.com`>, ajoutez la ligne reproduite ci-dessous au fichier `virtusertable`:

```
@example.com     bob@other-example.com
```

Pour finaliser cette modification, le fichier `virtusertable.db` doit être mis à jour à l'aide de la commande suivante, en étant connecté en tant que super-utilisateur:

```
makemap hash /etc/mail/virtusertable < /etc/mail/virtusertable
```

Ce faisant, un nouveau fichier `virtusertable.db` est créé, reflétant la nouvelle configuration.

11.3.1.3. Modifications courantes de la configuration de Sendmail

Lors de la modification du fichier de configuration Sendmail, il est recommandé de générer un tout nouveau un fichier `/etc/mail/sendmail.cf` plutôt que de modifier un fichier existant.



Attention

Avant de modifier le fichier `sendmail.cf`, il est toujours conseillé d'effectuer une copie de sauvegarde de la version courante du fichier.

Pour ajouter la fonctionnalité désirée à Sendmail, éditez le fichier `/etc/mail/sendmail.mc`. Une fois cette opération terminée, utilisez le macroprocesseur `m4` pour générer un nouveau fichier `sendmail.cf` en exécutant la commande `m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf`. Après la création d'un nouveau fichier `/etc/mail/sendmail.cf`, redémarrez Sendmail pour qu'il reflète les changements apportés. Pour ce faire, le moyen le plus simple consiste à taper la commande `/sbin/service sendmail restart`, en étant connecté en tant que super-utilisateur.

Par défaut, le macroprocesseur `m4` est installé avec Sendmail mais fait partie du paquetage `m4`.



Important

Le fichier `sendmail.cf` par défaut n'autorise pas Sendmail à accepter des connexions réseau de tout hôte autre que l'ordinateur local. Afin de configurer Sendmail en tant que serveur pour d'autres clients, éditez `/etc/mail/sendmail.mc` et modifiez les valeurs de `DAEMON_OPTIONS` pour permettre l'écoute des périphériques de réseau ou supprimez tout simplement les commentaires appropriés pour cette option. Régénérez ensuite le fichier `/etc/mail/sendmail.cf` grâce à la commande:

```
m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf
```

Cette configuration devrait fonctionner pour la plupart des sites exclusivement SMTP. Elle *ne fonctionnera pas* pour les sites UUCP ('UNIX to UNIX Copy'); vous devrez générer un nouveau fichier `sendmail.cf` si vous devez utiliser les transferts de courrier UUCP.

Consultez le fichier `/usr/share/sendmail-cf/README` avant de modifier tout fichier contenus dans les répertoires sous le répertoire `/usr/share/sendmail-cf`, car ils peuvent affecter la configuration future de fichiers `/etc/mail/sendmail.cf`.

11.3.1.4. Masquarade

L'une des configurations courantes de Sendmail est d'avoir un seul ordinateur qui agit comme passerelle de messagerie pour tous les ordinateurs sur un réseau. Par exemple, une société pourrait souhaiter qu'un ordinateur appelé `mail.bigcorp.com` gère tout son courrier électronique et attribue à tous les messages sortants la même adresse de retour.

Dans ce cas de figure, le serveur Sendmail est obligé de déguiser le nom des ordinateurs du réseau de la société de façon à ce que leur adresse de retour soit user@bigcorp.com au lieu de user@devel.bigcorp.com.

Pour ce faire, ajoutez les lignes suivantes à `/etc/mail/sendmail.mc`:

```
FEATURE(always_add_domain)dnl
FEATURE(`masquerade_entire_domain')
FEATURE(`masquerade_envelope')
FEATURE(`allmasquerade')
MASQUERADE_AS(`bigcorp.com.')
MASQUERADE_DOMAIN(`bigcorp.com.')
MASQUERADE_AS(bigcorp.com)
```

Une fois qu'elle aura généré un nouveau `sendmail.cf` à l'aide de `m4`, cette configuration fera en sorte que le tous les messages envoyés à partir du réseau semblent avoir été envoyés depuis `bigcorp.com`.

11.3.1.5. Blocage des spams

Les 'spams' (ou pourriel) peuvent être définis comme étant des messages électroniques inutiles et indésirables reçus par un utilisateur qui n'en a jamais fait la demande. Il s'agit d'un abus très perturbateur, coûteux et répandu des normes de communication Internet.

Sendmail rend relativement aisé le blocage des nouvelles techniques utilisées pour envoyer des spams. Il bloque même par défaut, un grand nombre des méthodes d'envoi de spams les plus courantes.

Par exemple, le réacheminement de messages SMTP, aussi appelé retransmission ('relaying'), a été désactivé par défaut depuis la version 8.9. de Sendmail. Auparavant, Sendmail aurait dirigé l'hôte de messagerie (`x.org`) de façon à ce qu'il accepte des messages d'un individu (`y.com`) et les envoie à un autre individu (`z.net`). Désormais, Sendmail doit être configuré de façon à autoriser un domaine à retransmettre du courrier par le biais du serveur. Pour configurer les domaines de retransmission, éditez simplement le fichier `/etc/mail/relay-domains` et relancez Sendmail.

Ceci étant, les utilisateurs sont très souvent bombardés de pourriel provenant d'autres serveurs via l'Internet. Dans ce cas, les fonctions de contrôle d'accès de Sendmail, disponibles par l'entremise du fichier `/etc/mail/access` peuvent servir à empêcher les connexions en provenance d'hôtes indésirables. L'exemple suivant illustre comment utiliser ce fichier pour non seulement bloquer mais également autoriser l'accès au serveur Sendmail:

```
badspammer.com        ERROR:550 "Go away and do not spam us anymore"
tux.badspammer.com    OK
10.0                  RELAY
```

Cet exemple stipule que tout message électronique envoyé par `badspammer.com` doit être bloqué à l'aide d'un code d'erreur 550 conforme à RFC-821 et qu'un message doit être renvoyé à l'expéditeur de pourriel. Le courrier envoyé par le sous-domaine `tux.badspammer.com` en revanche peut être accepté. La dernière ligne montre que tout message envoyé depuis le réseau 10.0.*.* peut être retransmis au moyen de votre serveur de messagerie.

Étant donné que `/etc/mail/access.db` est une base de données, vous devez utiliser `makemap` pour activer toute modification. Pour ce faire, tapez la commande suivante en étant connecté en tant que super-utilisateur:

```
makemap hash /etc/mail/access < /etc/mail/access
```

Comme vous pouvez l'imaginer, cet exemple ne fait qu'effleurer la surface du potentiel de Sendmail en termes d'autorisation ou d'interdiction d'accès. Reportez-vous au document `/usr/share/doc/sendmail/README.cf` pour obtenir de plus amples renseignements et d'autres exemples.

Étant donné que Sendmail fait appel à l'ADC Procmail pour la livraison de courrier, il est également possible d'utiliser un programme de filtrage de pourriel comme SpamAssassin, pour identifier et classer ce type de courrier à la place de l'utilisateur. Reportez-vous à la Section 11.4.2.6 pour obtenir de plus amples informations sur l'utilisation de du programme SpamAssassin.

11.3.1.6. Utilisation de Sendmail avec LDAP

L'utilisation de *Lightweight Directory Access Protocol (LDAP)* est une façon très rapide et puissante de trouver des informations spécifiques sur un utilisateur particulier appartenant à un grand groupe. Par exemple, un serveur LDAP peut servir à chercher une adresse électronique spécifique dans un répertoire d'entreprise à partir du nom de famille de l'utilisateur. Pour ce genre de mise en application, LDAP est en grande partie séparé de Sendmail; LDAP stocke les informations hiérarchiques des utilisateurs alors que Sendmail ne s'occupe que de recevoir le résultat de la recherche LDAP par le biais de messages électroniques pré-adressés.

Toutefois, Sendmail prend en charge une intégration beaucoup plus grande avec LDAP, là où il utilise LDAP pour remplacer des fichiers maintenus séparément, tels que `aliases` et `virtusertables`, sur divers serveurs de messagerie qui fonctionnent ensemble pour prendre en charge une organisation de taille moyenne ou supérieure. En bref, LDAP fait abstraction du niveau de routage du courrier depuis Sendmail et ses fichiers de configuration séparés en un cluster LDAP puissant qui influence de nombreuses autres applications.

La version actuelle de Sendmail comprend la prise en charge pour LDAP. Pour étendre votre serveur Sendmail à l'aide de LDAP, prenez d'abord un serveur LDAP, tel que **OpenLDAP**, opérationnel et correctement configuré. Ensuite, modifiez votre fichier `/etc/mail/sendmail.mc` pour y inclure les éléments suivants:

```
LDAPROUTE_DOMAIN('yourdomain.com')dnl
FEATURE('ldap_routing')dnl
```



Remarque

Ceci n'est que pour une configuration de base de Sendmail avec LDAP. Votre configuration devrait différer considérablement de celle-ci selon votre mise en application de LDAP, tout spécialement si vous souhaitez configurer plusieurs ordinateurs Sendmail pour qu'ils utilisent un serveur LDAP commun.

Consultez `/usr/share/doc/sendmail/README.cf` pour avoir des informations de configuration de routage LDAP détaillées et des exemples.

Ensuite, recréez le fichier `/etc/mail/sendmail.cf` en exécutant `m4` et redémarrant Sendmail. Reportez-vous à la Section 11.3.1.3 pour obtenir des instructions sur la manière de procéder.

Pour plus d'informations sur LDAP, reportez-vous au Chapitre 13.

11.3.2. Fetchmail

Fetchmail est un ATC récupérant du courrier électronique depuis des serveurs distants et le transfère à l'ATC local. De nombreux utilisateurs apprécient le fait de pouvoir séparer le processus de téléchargement de leurs messages stockés sur un serveur distant, du processus de lecture et d'organisation de leur courrier dans un AGC. Conçu tout spécialement pour les utilisateurs qui se connectent par modem, Fetchmail se connecte et télécharge rapidement tous les messages électroniques dans le fichier spoule de messagerie à l'aide de nombreux protocoles différents, tels que POP3 et IMAP. Il permet même de réacheminer vos messages vers un serveur SMTP, si nécessaire.

Fetchmail est configrué pour chaque utilisateur grâce à un fichier `.fetchmailrc` du répertoire personnel de l'utilisateur.

Sur la base des préférences spécifiés dans le fichier `.fetchmailrc`, Fetchmail recherche les messages électroniques sur un serveur distant et les récupère. Il essaie ensuite de les acheminer au port 25 de l'ordinateur local, au moyen de l'ATC local, pour placer les messages sur le fichier spoule de l'utilisateur approprié. Si Procmail est disponible, il peut ensuite être utilisé pour filtrer les messages et les placer dans une boîte à lettres de sorte qu'ils puissent être lus avec un AGC.

11.3.2.1. Options de configuration de Fetchmail

Bien qu'il soit possible de passer toutes les options nécessaires pour vérifier le courrier sur un serveur distant depuis la ligne de commande lorsque l'on exécute Fetchmail, il est beaucoup plus simple d'utiliser un fichier `.fetchmailrc`. Toutes les options de configuration vont certes dans le fichier `.fetchmailrc` mais il est possible de les écraser lorsque Fetchmail est en cours en spécifiant cette option à la ligne de commande.

Le fichier `.fetchmailrc` d'un utilisateur est divisé en trois types d'option de configuration:

- *options globales* — donne à Fetchmail des instructions qui contrôlent l'exploitation du programme ou fournit des réglages pour toute connexion de vérification du courrier.
- *options serveur* — Spécifie les informations nécessaires sur le serveur scruté, telles que le nom d'hôte, de même que les préférences que vous souhaitez utiliser avec un serveur de messagerie donné, comme le port à vérifier ou le nombre de secondes d'attente avant d'interrompre la connexion. Ces options affectent chaque option utilisateur utilisée avec ce serveur.
- *options utilisateur* — Contient des informations, telles que le nom d'utilisateur et le mot de passe, nécessaires à l'authentification et la vérification du courrier à l'aide d'un serveur de messagerie donné.

Les options globales apparaissent au sommet du fichier de configuration `.fetchmailrc`, suivies d'une ou plusieurs options serveur, précisant chacune un serveur de messagerie différent sur lequel Fetchmail devrait vérifier le courrier. Les options utilisateur vont à la suite des options serveur pour chaque compte utilisateur devant être vérifié sur ce serveur de messagerie. Tout comme les options serveur, il est possible de spécifier non seulement de multiples options utilisateur à utiliser avec un serveur donné mais également de vérifier plusieurs comptes de courrier sur un même serveur.

Les options serveur sont appelées à être utilisées dans le fichier `.fetchmailrc` par l'emploi d'un verbe d'option spécial, `poll` ou `skip`, qui précède toute information serveur. L'action `poll` indique à Fetchmail d'utiliser cette option serveur lorsqu'il est exécuté; il vérifie en fait le courrier à l'aide

des différentes options utilisateur. Toute option serveur après une action `skip`, n'est pas vérifiée, à moins que le nom d'hôte de ce serveur ne soit spécifié lorsque Fetchmail est invoqué. L'option `skip` établit des configurations test dans `.fetchmailrc` et ne vérifie ce serveur que selon des instructions spécifiques, sans affecter toute autre configuration actuellement en cours.

Ci-dessous figure un exemple de fichier `.fetchmailrc`:

```
set postmaster "user1"
set bouncemail

poll pop.domain.com proto pop3
     user 'user1' there with password 'secret' is user1 here

poll mail.domain2.com
     user 'user5' there with password 'secret2' is user1 here
     user 'user7' there with password 'secret3' is user1 here
```

Dans cet exemple, les options globales sont configurées de façon à ce que l'utilisateur reçoive le courrier seulement en dernier ressort (option `postmaster`) et que toutes les erreurs soient envoyées au "postmaster" plutôt qu'à l'expéditeur (option `bouncemail`). L'action `set` indique à Fetchmail que cette ligne contient une option globale. Ensuite, deux serveurs de messagerie sont spécifiés; le premier, configuré pour vérifier POP3 et le second pour essayer divers protocoles afin d'en trouver un qui fonctionne. Deux utilisateurs sont vérifiés dans le cas de la seconde option serveur, mais tout message électronique trouvé pour l'un ou l'autre des utilisateurs est envoyé dans le fichier spoule de messagerie de l'utilisateur 1. Ceci permet de vérifier des boîtes à lettres multiples sur des serveurs multiples, bien qu'apparaissant dans un seul AGC. Chaque information spécifique à un utilisateur commence par l'action `user`.



Remarque

Les utilisateurs ne doivent pas placer leur mot de passe dans le fichier `.fetchmailrc`. Si la section `with password '`*<mot-de-passe>*`'` est omise Fetchmail demandera un mot de passe lors de son lancement.

Fetchmail offre de nombreuses options différentes, tant globales que serveur ou locales. Un grand nombre de ces options sont rarement utilisées ou ne s'appliquent qu'à des situations très particulières. La page de manuel relative à `fetchmail` explique chacune de ces options de façon détaillée, mais les options les plus courantes sont énumérées ci-dessous.

11.3.2.2. Options globales

Chaque option globale devrait être placée sur une ligne individuelle et précédée de l'action `set`.

- `daemon` *<seconds>* — Spécifie le mode démon dans lequel Fetchmail demeure en tâche de fond et récupère le courrier à intervalle déterminé.
- `postmaster` — Spécifie un utilisateur local auquel envoyer le courrier en cas de problèmes de distribution.
- `syslog` — Spécifie le fichier journal pour l'enregistrement des messages d'erreur et d'état. Par défaut, `/var/log/maillog` est retenu.

11.3.2.3. Options serveur

Les options serveur doivent figurer sur leur propre ligne dans `.fetchmailrc`, après une action `poll` ou `skip`.

- `auth` <*type-d'auth-*> — Spécifie le type d'authentification à utiliser. Par défaut, l'authentification `password` est utilisée, mais certains protocoles prennent en charge d'autres types d'authentification, notamment `kerberos_v5`, `kerberos_v4` et `ssh`. Si le type d'authentification `any` est retenu, Fetchmail essaiera d'abord des méthodes qui ne nécessitent aucun mot de passe, puis des méthodes qui masquent votre mot de passe et, en dernier ressort, essaiera d'envoyer votre mot de passe en texte en clair pour effectuer l'authentification au serveur.
- `interval` <*nombre*> — Indique à Fetchmail de ne scruter que ce serveur chaque <*nombre*> de fois qu'il vérifie le courrier sur tous les serveurs configurés. Cette option est généralement utilisée pour les serveurs de messagerie sur lesquels un utilisateur ne reçoit que peu de messages.
- `port` <*numéro-de-port*> — Écrase le numéro de port par défaut pour un protocole spécifié.
- `proto` <*protocole*> — Spécifie un protocole particulier, tel que `pop3` ou `imap`, à utiliser pour vérifier le courrier sur ce serveur.
- `timeout` <*secondes*> — Spécifie la durée d'inactivité du serveur)en secondes) après laquelle Fetchmail abandonne une tentative de connexion. Si cette valeur n'est pas configurée, le système retient une valeur par défaut de `300` secondes.

11.3.2.4. Options utilisateur

Les options utilisateur peuvent être placées sur leurs propres lignes sous une option serveur ou alors sur la même ligne qu'une option serveur. Dans les deux cas, les options utilisateur doivent suivre l'option `user` (définie ci-dessous).

- `fetchall` — Donne l'ordre à Fetchmail de télécharger tous les messages d'une file, y compris les messages qui ont déjà été visualisés. Par défaut, Fetchmail ne récupère que les nouveaux messages.
- `fetchlimit` <*nombre*> — Ne permet le téléchargement que d'un certain nombre de messages avant l'arrêt.
- `flush` — Donne l'instruction à Fetchmail de supprimer tous les messages de la file visualisés précédemment avant de télécharger les nouveaux messages.
- `limit` <*nombre-max-octets*> — Spécifie que seuls les messages dont la taille est inférieure à la taille spécifiée peuvent être récupérés. Cette option est pratique lors de connexions réseau lentes, particulièrement lorsqu'un gros message prend trop de temps à télécharger.
- `password '`<*mot de passe*>`'` — Spécifie le mot de passe à utiliser pour cet utilisateur.
- `preconnect "`<*commande*>`"` — Exécute la commande spécifiée avant de récupérer les messages pour cet utilisateur.
- `postconnect "`<*commande*>`"` — Exécute la commande spécifiée après avoir récupéré les messages pour cet utilisateur.
- `ssl` — Active le cryptage SSL.
- `user "`<*nom-d'utilisateur*>`"` — Définit le nom d'utilisateur employé par Fetchmail pour récupérer les messages électroniques. *Cette option doit être placée avant toute autre option utilisateur.*

11.3.2.5. Options de commande Fetchmail

La plupart des options utilisées à la ligne de commande lors de l'exécution de la commande `fetchmail`, répliquent les options de configuration de `.fetchmailrc`. Ainsi, Fetchmail peut être utilisé avec ou sans fichier de configuration. La plupart des utilisateurs n'utilisent jamais ces options à la ligne de commande car il est plus simple de les laisser dans le fichier `.fetchmailrc` et de les utiliser chaque fois que Fetchmail est exécuté.

Toutefois, il se peut que dans certaines situations, la commande `fetchmail` doive être exécutée avec d'autres options dans un but bien précis. Étant donné que les options spécifiées à la ligne de commande écrasent les options du fichier de configuration, il est possible d'exécuter des options de commande pour écraser temporairement un paramétrage de `.fetchmailrc` qui est la cause d'une erreur.

11.3.2.6. Options d'information ou de débogage

Certaines options utilisées après la commande `fetchmail` permettent d'obtenir d'importantes informations.

- `--configdump` — Affiche toutes les options possibles sur la base des informations de `.fetchmailrc` et les valeurs par défaut de Fetchmail. Aucun message électronique n'est téléchargé lorsque vous utilisez cette option, et ce, pour aucun utilisateur.
- `-s` — Exécute Fetchmail en mode silencieux, empêchant tout message, autre que des messages d'erreur, d'apparaître après la commande `fetchmail`.
- `-v` — Exécute Fetchmail en mode prolixe, affichant toute communication entre Fetchmail et les serveurs de messagerie distants.
- `-V` — Affiche des informations détaillées sur la version utilisée, la liste des options globales et les paramètres à appliquer à chaque utilisateur, y compris le protocole de messagerie et la méthode d'authentification. Lors de l'utilisation de cette option, aucun courrier électronique n'est récupéré pour quelque utilisateur que ce soit.

11.3.2.7. Options spéciales

Ces options peuvent parfois être pratiques pour écraser les valeurs par défaut qui se trouvent souvent dans le fichier `.fetchmailrc`.

- `-a` — Indique à Fetchmail de télécharger tous les messages depuis le serveur de messagerie distant, qu'ils soient nouveaux ou déjà visualisés. Par défaut, Fetchmail ne télécharge que les nouveaux messages.
- `-k` — Fait en sorte que Fetchmail laisse les messages sur le serveur de messagerie distant après les avoir téléchargés. Cette option écrase le comportement par défaut qui consiste à supprimer les messages après les avoir téléchargés.
- `-l` <*nombre-max-octets*> — Indique à Fetchmail de ne pas télécharger les messages dont la taille est supérieure à la taille spécifiée et de les laisser sur le serveur de messagerie distant.
- `--quit` — Quitte le processus démon de Fetchmail.

D'autres commandes et options `.fetchmailrc` sont disponibles dans la page de manuel relative à `fetchmail`.

11.4. Agent de distribution de courrier (ADC)

Red Hat Linux inclut deux ADC primaire, à savoir Procmail et `mail`. Ces deux applications sont considérées comme des agents de distribution de courrier (ou ADC) locaux et toutes les deux transmettent le courrier électronique du fichier spoule d'un ADC à la boîte à lettres de l'utilisateur. Toutefois, Procmail fournit un système de filtrage robuste.

Cette section examine seulement Procmail de façon détaillée. Pour toute information sur la commande `mail`, consultez la page de manuel qui lui est dédié.

Procmail distribue et filtre le courrier électronique du moment où il est placé dans le fichier spoule de messagerie de l'hôte local. Il est puissant, peu exigeant en matière de ressources de système et très utilisé. Procmail peut jouer un rôle critique dans la distribution du courrier qui sera lu par les applications client de messagerie.

Il existe différentes façons d'invoquer Procmail. Dès qu'un ACT dépose un message dans le fichier spoule de messagerie Procmail est lancé. Ce dernier filtre et classe le courrier de manière à ce que l'ACT puisse le trouver et quitte. L'ACT peut également être configuré de sorte qu'il exécute Procmail chaque fois qu'un message est reçu afin que le courrier soit acheminé vers les boîtes à lettres appropriées. Par défaut, la présence d'un fichier `.procmailrc` dans le répertoire personnel d'un utilisateur invoquera Procmail dès qu'un ACT reçoit un nouveau message.

Les actions effectuées sur un message électronique par Procmail dépendent des instructions de *recettes* particulières, ou règles, par rapport auxquelles les messages sont comparés. Si un message correspond à la recette, il peut alors être placé dans un fichier donné, supprimé ou traité d'une autre façon.

Lorsque Procmail est lancé, il lit les messages électroniques et sépare le corps de message des informations d'en-tête. Ensuite, Procmail cherche les fichiers `/etc/procmailrc` et `rc` dans le répertoire `/etc/procmailrcs` pour trouver les variables d'environnement Procmail, d'ensemble et par défaut, ainsi que les recettes. Procmail cherche alors un fichier `.procmailrc` dans le répertoire personnel de l'utilisateur pour trouver des règles spécifiques à cet utilisateur. De nombreux utilisateurs créent des fichiers `rc` supplémentaires pour Procmail, qui sont référencés par leur fichier `.procmailrc` mais peuvent être activés ou désactivés rapidement en cas de problème lors de la filtration de messages.

Par défaut, aucun fichier `rc` pour l'ensemble du système n'existe dans le répertoire `/etc` et aucun fichier utilisateur `.procmailrc` n'existe. Pour commencer à utiliser Procmail, créez un fichier `.procmailrc` contenant des variables d'environnement et des règles spécifiques pour certains types de messages.

Dans la plupart des configurations, la décision de lancer Procmail et de tenter de filtrer le courrier est basée sur l'existence d'un fichier utilisateur `.procmailrc`. Pour désactiver Procmail, mais enregistrer votre travail dans le fichier `.procmailrc`, déplacez-le vers un nom de fichier similaire à l'aide de la commande `mv ~/.procmailrc ~/.procmailrcSAVE`. Lorsque vous êtes prêt à tester Procmail de nouveau, redonnez au fichier son nom original, soit `.procmailrc`. Procmail recommencera à fonctionner immédiatement.

11.4.1. Configuration de Procmail

Fichiers de configuration de Procmail, mieux connus comme étant les fichiers utilisateur `.procmailrc`, contiennent d'importantes variables d'environnement. Celles-ci indiquent à Procmail quels messages trier, quoi faire avec les messages qui ne correspondent à aucune recette, etc.

Ces variables d'environnement se trouvent généralement au début du fichier `.procmailrc` au format suivant:

```
<variable-env>="<valeur>"
```

Dans cet exemple, <*variable-env*> représente le nom de la variable et la section <*valeur*> définit la variable.

La plupart des utilisateurs de Procmail se servent d'un petit nombre de variables et la plupart des variables d'environnement les plus importantes sont déjà définies à l'aide d'une valeur par défaut. Généralement, les variables suivantes seront utilisées:

- `DEFAULT` — Définit la boîte à lettres où seront placés les messages qui ne correspondent à aucune recette.

  La valeur `DEFAULT` par défaut est la même que `$ORGMAIL`.

- `INCLUDERC` — Spécifie des fichiers `rc` supplémentaires qui contiennent d'autres recettes servant à comparer les messages. Ceci permet de diviser vos listes de recettes Procmail en fichiers individuels qui jouent différents rôles, tels que le blocage de spams et la gestion de listes d'adresses électroniques, qui peuvent ensuite être activés ou désactivés à l'aide de caractères de commentaire dans le fichier `.procmailrc` de l'utilisateur.

  Par exemple, des lignes dans un fichier `.procmailrc` de l'utilisateur peuvent ressembler à l'extrait suivant:

  ```
  MAILDIR=$HOME/Msgs
  INCLUDERC=$MAILDIR/lists.rc
  INCLUDERC=$MAILDIR/spam.rc
  ```

  Si l'utilisateur souhaite désactiver la filtration Procmail de ses listes d'adresses, mais désire laisser le contrôle des spams en place, il n'a qu'à commenter la première ligne `INCLUDERC` avec le symbole dièse ([#]).

- `LOCKSLEEP` — Définit la durée, en secondes, entre les tentatives de Procmail d'utiliser un fichier de verrouillage donné. La valeur par défaut est 8 secondes.

- `LOCKTIMEOUT` — Définit la durée, en secondes, qui doit s'écouler après la dernière modification d'un fichier de verrouillage avant que Procmail ne considère le fichier de verrouillage comme étant vieux et pouvant par conséquent être supprimé. La valeur par défaut est 1024 secondes.

- `LOGFILE` — L'emplacement et le fichier devant contenir tout message d'erreur ou d'information Procmail.

- `MAILDIR` — Règle le répertoire de travail en cours pour Procmail. S'il est réglé, tous les autres chemins Procmail sont relatifs à ce répertoire.

- `ORGMAIL` — Spécifie la boîte à lettres originale ou un autre endroit où placer les messages s'ils ne peuvent être placés à l'emplacement par défaut ou requis par la recette.

  Par défaut, une valeur de `/var/spool/mail/$LOGNAME` est utilisée.

- `SUSPEND` — Définit la durée, en secondes, de pause de Procmail si une ressource nécessaire, telle que l'espace swap, n'est pas disponible.

- `SWITCHRC` — Permet à un utilisateur de spécifier un fichier externe contenant des recettes Procmail supplémentaires; plus ou moins comme l'option `INCLUDERC`, sauf que la vérification des recettes est arrêtée sur le fichier de configuration traitant et seules les recettes sur le fichier spécifié avec `SWITCHRC` sont utilisées.

- `VERBOSE` — Fait en sorte que Procmail journalise beaucoup plus d'informations. Cette option est pratique pour le débogage.

D'autres variables d'environnement importantes sont obtenues depuis le shell, comme `LOGNAME`, qui est le nom de connexion; `HOME`, qui est l'emplacement du répertoire personnel; et `SHELL`, qui est le shell par défaut.

Consultez la page de manuel relative à `procmailrc` pour obtenir des explications exhaustives sur les variables d'environnement, de même que leurs valeurs par défaut.

11.4.2. Recettes Procmail

Les nouveaux utilisateurs trouvent généralement que les recettes constituent l'élément le plus difficile de l'apprentissage d'utilisation de Procmail. Ce sentiment est compréhensible, jusqu'à un certain point, étant donné que les recettes procèdent à la comparaison avec les messages à l'aide d'*expressions régulières*, qui est un format particulier utilisé pour spécifier des qualifications de concordance de chaînes. Ceci étant, les expressions régulières ne sont pas très compliquées à créer et le sont encore moins à comprendre et à lire. De plus, la cohérence avec laquelle les recettes Procmail sont écrites, sans tenir compte des expressions régulières, permet de comprendre facilement ce qui se passe.

L'explication exhaustive des expressions régulières va au-delà de la portée de ce chapitre. La structure des recettes Procmail est plus importante et des exemples pratiques de recettes Procmail figurent à différents endroits sur Internet (notamment http://www.iki.fi/era/procmail/links.html). Le bon usage et l'adaptation des expressions régulières qui se trouvent dans ces exemples de recettes dépendent de la compréhension de la structure des recettes Procmail. Des informations d'introduction spécifiques aux règles d'expressions régulières de base se trouvent dans la page de manuel relative à `grep` man page.

Une recette Procmail a la forme suivante:

```
:0<indicateurs>: <nom-fichier-verrouillage>

* <caractère-condition-spéciale>
<condition-1>
* <caractère-condition-spéciale>
<condition-2>
* <caractère-condition-spéciale>
<condition-N>

<caractère-action-spéciale><action-à-exécuter>
```

Les deux premiers caractères d'une recette Procmail sont le symbole des deux-points et un zéro. Divers indicateurs (flags) peuvent être placés après le zéro pour contrôler ce que fait Procmail lors du traitement de cette recette. Un deux-points placé après la section *<indicateurs>* spécifie qu'un fichier de verrouillage sera créé pour ce message. Si un fichier de verrouillage doit être créé, spécifiez son nom dans l'espace *<nom-fichier-verrouillage>*.

Une recette peut contenir plusieurs conditions servant à vérifier la concordance d'un message. S'il n'y a aucune condition, tous les messages auront une concordance positive avec la recette. Les expressions régulières sont placées dans certaines conditions de façon à faciliter la concordance avec les messages. Si l'on utilise des conditions multiples, elles doivent toutes obtenir la concordance pour qu'une action soit exécutée. Les conditions sont vérifiées sur la base des indicateurs spécifiés à la première ligne de la recette. Des caractères spéciaux facultatifs placés après le caractère `*` permettent de contrôler ultérieurement la condition.

<action-à-exécuter> spécifie ce qui arrive aux messages qui correspondent à l'une des conditions. Il ne peut y avoir qu'une action par recette. Dans de nombreux cas, le nom d'une boîte à lettres est utilisé à cet endroit pour envoyer les messages dans ce fichier, ce qui permet en fait de trier le courrier. Des caractères d'action spéciale peuvent également être utilisés avant que l'action ne soit spécifiée.

11.4.2.1. Recettes de distribution et de non-distribution

L'action utilisée si la recette correspond à un message donné détermine si la recette est considérée comme étant de distribution ou de non-distribution. Une *recette de distribution* contient une action qui écrit le message dans un fichier, envoie le message à un autre programme ou réachemine le message vers une autre adresse électronique. Une *recette de non-distribution* couvre toutes les autres actions, telles que l'utilisation d'un bloc d'imbrication. Un *bloc d'imbrication* est une action contenue entre accolades (`{ }`) et désignant des actions supplémentaires devant être exécutées sur les messages qui

correspondent aux conditions de la recette. Les blocs d'imbrication peuvent être emboîtés, offrant ainsi plus de contrôle pour l'identification et l'exécution d'actions sur les messages.

Les recettes de distribution qui correspondent à des messages font en sorte que Procmail exécute l'action spécifiée et cesse de comparer les messages en question aux autres recettes. Les messages qui correspondent aux conditions de recettes de non-distribution continuent d'être comparés aux autres recettes dans les fichiers `rc` courants et suivants. En d'autres termes, les recettes de non-distribution font en sorte que les messages continuent vers les autres recettes après l'exécution d'une action sur eux.

11.4.2.2. Indicateurs

Les indicateurs sont très importants pour déterminer la façon dont les conditions d'une recette sont comparées à un message et pour décider si elles doivent l'être ou non. Les indicateurs suivants sont couramment utilisés:

- `A` — Spécifie que cette recette ne sera utilisée que si la recette précédente sans indicateur `A` ou `a` a également obtenu la concordance avec ce message.

  Pour vous assurer que l'action sur cette dernière recette précédente correspondante a bel et bien été complétée avant d'accorder la concordance à la recette actuelle, utilisez plutôt l'indicateur `a`.
- `B` — Analyse le corps du message et recherche des conditions de concordance.
- `b` — Utilise le corps de message pour toute action résultante, telle que l'écriture du message dans un fichier ou son réacheminement. Il s'agit du comportement par défaut.
- `c` — Génère une copie conforme du message électronique. Ceci peut être pratique avec les recettes de distribution, étant donné que l'action requise peut être exécutée sur le message et que la copie du message peut continuer d'être traitée dans les fichiers `rc` files.
- `D` — Rend la comparaison `egrep` sensible à la casse. Par défaut, le processus de comparaison n'est pas sensible à la casse.
- `E` — Semblable à l'indicateur `A` sauf que les conditions dans cette recette ne sont comparées aux messages que si la recette immédiatement précédente sans indicateur `E` n'a pas obtenu la concordance. Cette action ressemble à une action *else*.

  Utilisez l'indicateur `e` si cette recette est vérifiée uniquement lorsque la recette précédente a obtenu une concordance, mais que l'action a échoué.
- `f` — Utilise le tube comme filtre.
- `H` — Analyse l'en-tête du message et recherche des conditions de concordance. Cela se fait par défaut.
- `h` — Utilise l'en-tête dans une action résultante. Cela est le comportement par défaut.
- `w` — Indique à Procmail d'attendre que le filtre ou le programme spécifiés aient terminé leurs opérations et fait son rapport, que l'opération précédente soit réussie ou non, avant de considérer le message comme étant filtré.

  Si vous voulez ignorer les messages "Program failure" lors de la décision du succès d'un filtre ou d'une action, utilisez l'option `W` à la place.

D'autres indicateurs sont expliqués dans la page de manuel `procmailrc`.

11.4.2.3. Spécification d'un fichier de verrouillage local

Les fichiers de verrouillage sont très utiles avec Procmail pour garantir que seul un processus qui essaie de modifier un certain message à un moment donné. Vous pouvez spécifier un fichier de verrouillage local en plaçant le symbole des deux points (`:`) après chaque indicateur sur la première ligne

d'une recette. Ce faisant, un fichier de verrouillage local est créé en fonction du nom de fichier de destination et de toute valeur contenue dans la variable d'environnement globale `LOCKEXT`.

Vous pouvez aussi spécifier le nom du fichier de verrouillage local à utiliser avec cette recette après le symbole des deux points (:).

11.4.2.4. Conditions et actions spéciales

Des caractères particuliers utilisés devant les conditions et les actions des recettes Procmail modifient la façon dont elles sont interprétées.

Les caractères suivants peuvent être utilisés après le symbole `*`, au début d'une ligne de condition d'une recette:

- `!` — Dans la ligne de condition, ce caractère inverse la condition, de sorte que la concordance ne sera désormais établie que si la condition ne correspond pas au message.
- < — Vérifie si la taille du message est inférieure au nombre d'octets spécifié.
- > — Vérifie si la taille du message est supérieure au nombre d'octets spécifié.

Les caractères suivants sont utilisés pour exécuter des actions spéciales:

- `!` — Dans la ligne d'action, ce caractère indique à Procmail de réacheminer le message vers les adresses électroniques spécifiées.
- `$` — Renvoie à une variable réglée précédemment dans le fichier `rc`. Ceci est généralement utilisé pour définir une boîte à lettres commune à laquelle diverses recettes feront référence.
- `|` — Le caractère de tube indique à Procmail de lancer un programme spécifique pour traiter ce message.
- `{` and `}` — Construit un bloc d'imbrication, utilisé pour contenir des recettes supplémentaires à appliquer aux messages comparés.

Si aucun caractère spécial n'est utilisé au début de la ligne d'action, Procmail considère alors que la ligne d'action spécifie une boîte à lettres où les messages devraient être écrits.

11.4.2.5. Exemples de recettes

Procmail est un programme extrêmement flexible, vous permettant de comparer des messages sur la base de conditions très spécifiques et ensuite d'exécuter des actions détaillées sur ces derniers. Toutefois, pour les nouveaux utilisateurs, cette flexibilité peut rendre difficile la création d'une recette Procmail de toutes pièces visant à atteindre un objectif bien précis.

La meilleure façon de développer vos aptitudes en matière de création de recettes Procmail consiste à bien comprendre les expressions régulières et à examiner attentivement de nombreux exemples de recettes créées par d'autres utilisateurs. Les quelques exemples simples suivants ont pour but de vous montrer la structure des recettes Procmail et peuvent servir de base pour la construction de structures plus complexes.

Une recette élémentaire ne contient pas forcément de conditions, comme le montre l'exemple ci-dessous:

```
:0:
new-mail.spool
```

La première ligne commence la recette en spécifiant qu'un fichier de verrouillage local doit être créé, mais n'indique aucun nom, laissant Procmail utiliser le nom de fichier de la destination et `LOCKEXT` le nommer. Étant donné qu'aucune condition n'est spécifiée, tous les messages correspondent à cette

recette et sont par conséquent placés dans le fichier spoule unique appelé `new-mail.spool`, situé dans le répertoire spécifié par la variable d'environnement `MAILDIR`. Un AGC peut ensuite visualiser les messages dans ce fichier.

Cette recette de base pourrait être placée à la fin de tous les fichiers `rc` afin d'acheminer les messages vers un emplacement par défaut. Un exemple plus complexe pourrait prendre des messages provenant d'une adresse électronique donnée et les supprimer, comme le montre ce exemple.

```
:0
* ^From: spammer@domain.com
/dev/null
```

Dans le cas de cet exemple, tout message envoyé par `spammer@domain.com` est automatiquement déplacé vers `/dev/null`, qui le supprime.



Attention

Soyez très prudent lorsque vous effectuez ce genre d'opération et assurez-vous que la règle fonctionne correctement avant de déplacer les messages qui y correspondent vers `/dev/null`, car ils y seront supprimés de façon permanente. Si les conditions de votre recette attrapent accidentellement des messages qui ne devraient pas l'être, ils disparaissent sans laisser de trace. Dans de telles conditions, il est difficile de résoudre des problèmes au niveau de la règle.

Une solution plus appropriée serait de pointer l'action de la recette vers une boîte aux lettres spéciale que vous pouvez vérifier de temps en temps, afin de voir s'il s'y trouve de *fausses concordances* ou des messages qui correspondent par accident aux conditions. Après un examen méticuleux donnant l'assurance qu'aucun message ne fait l'objet d'une concordance accidentelle, supprimer la boîte à lettres et diriger l'action de façon à envoyer les messages vers `/dev/null`.

Procmail est avant tout un filtre de courrier électronique, qui place automatiquement le courrier au bon endroit pour vous éviter de le trier manuellement. La recette ci-dessous prend les messages envoyés par une liste d'adresses donnée et les met dans le dossier approprié.

```
:0:
* ^(From|CC|To).*tux-lug
tuxlug
```

Tout message envoyé depuis la liste d'adresses `tux-lug@domain.com` sera automatiquement placé dans la boîte à lettres `tuxlug` pour le AGC. Notez que la condition dans cet exemple permettra d'obtenir une concordance avec des messages si des adresses électroniques de la liste d'adresses se trouvent sur l'une des lignes suivantes: `From`, `CC` ou `To`.

Pour obtenir des informations sur des recettes plus détaillées et puissantes, consultez l'une des nombreuses ressources en ligne sur Procmail disponibles dans la Section 11.6.

11.4.2.6. Filtres de spam

Puisque Procmail est appelé par Sendmail, Postfix et Fetchmail lors de la réception de nouveaux messages, il peut être utilisé comme un outil puissant pour combattre le pourriel.

Le combat contre le pourriel est encore plus efficace lorsque Procmail est utilisé de concert avec SpamAssassin. En effet, grâce à une double action ces deux applications peuvent rapidement identifier des messages-pourriel, de les trier et de les détruire.

SpamAssassin recours à une analyse de l'en-tête et du texte, à des listes noires et à des bases de données de localisation de spam pour identifier et étiqueter tout pourriel.

Pour un utilisateur local, la meilleure façon d'utiliser SpamAssassin consiste à insérer la ligne suivante vers le haut du fichier `~/.procmailrc`:

```
INCLUDERC=/etc/mail/spamassassin/spamassassin-default.rc
```

Le programme `/etc/mail/spamassassin/spamassassin-default.rc` contient une simple règle Procmail permettant d'activer SpamAssassin pour tout courrier électronique reçu. Si un message est reconnu comme étant un pourriel, il est étiqueté en tant que tel dans l'en-tête et le titre se voit inclure la mention suivante:

```
*****SPAM*****
```

Le corps du message de l'email est précédé d'un compte rendu des éléments ayant justifié le diagnostic de spam.

Pour classer les emails étiquetés en tant que pourriel, il est possible d'utiliser une règle semblable à celle reproduite ci-dessous:

```
:0 Hw
* ^X-Spam-Status: Yes
spam
```

Selon cette règle, tous les messages étiquetés en tant que spam dans l'en-tête sont rangés dans une boîte à lettres nommée `spam`.

Étant donne que SpamAssassin est un script Perl, il faudra peut-être nécessaire, d'utiliser le démon binaire SpamAssassin (`spamd`) et l'application client (`spamc`) sur les serveur très sollicités. Pour configurer SpamAssassin de la sorte, l'accès super-utilisateur à l'hôte est nécessaire.

Pour lancer le démon `spamd`, tapez la commande suivante en étant connecté en tant que super-utilisateur (ou root):

```
/sbin/service spamassassin start
```

Pour que le démon SpamAssassin puisse être lancé lors du démarrage du système, utilisez un utilitaire 'initscript', comme l'**Outil de configuration des services** (`redhat-config-services`), pour activer le service `spamassassin`. Reportez-vous à la Section 1.4.2 pour de plus amples informations sur les utilitaires initscript.

Pour configurer Procmail afin qu'il utilise l'application client SpamAssassin au lieu du script Perl, placez le ligne suivante vers le haut du fichier `~/.procmailrc` ou, pour une configuration du système en général, placez-la dans `/etc/procmailrc`:

```
INCLUDERC=/etc/mail/spamassassin/spamassassin-spamc.rc
```

11.5. Agent de gestion de courrier (AGC)

De nombreux progammes de messagerie sont disponibles sous Red Hat Linux. Parmi eux figurent des programmes de messagerie client graphique dotés de nombreuses fonctions comme **Mozilla Mail** ou **Ximian Evolution**, ainsi que des programmes de messagerie à base de texte comme **mutt** ou **pine**.

Pour obtenir des informations sur l'utilisation de ces applications, reportez-vous au chapitre intitulé *Applications de messagerie* du *Guide de démarrage de Red Hat Linux*.

Le reste de cette section se concentre sur l'établissement d'une communication sécurisée entre le client et le serveur.

11.5.1. Établissement d'une communication sécurisée

Les AGC très utilisés fournis avec Red Hat Linux, tels que **Mozilla Mail**, `mutt` et `Pine` offrent des sessions de courrier électronique cryptées avec SSL.

Comme pour tout autre service voyageant sur un réseau non-crypté, des informations de messagerie importantes comme les noms d'utilisateur, mots de passe et des messages entiers, peuvent être interceptées et lues par des utilisateurs du réseau. En outre, étant donné que les protocoles POP et IMAP standard transfèrent les informations d'authentification en texte clair, un pirate peut obtenir l'accès aux comptes utilisateur en collectionnant les noms d'utilisateur et mots de passe alors qu'ils sont transférés sur le réseau.

11.5.1.1. Clients de messagerie sécurisés

Heureusement, la plupart des AGC Linux conçus pour vérifier le courrier sur des serveurs distants prennent en charge le cryptage SSL. Afin de pouvoir utiliser SSL lors de la récupération du courrier, il doit être activé aussi bien sur le client de messagerie que sur le serveur de messagerie.

SSL est généralement très simple à activer du côté client, il suffit même parfois de cliquer sur un bouton dans la fenêtre de configuration de l'AGC ou de l'activer au moyen d'une option dans le fichier de configuration de l'AGC. Les IMAP et POP sécurisés ont des numéros de port connus (993 et 995 respectivement) que l'AGC utilise pour authentifier et télécharger les messages.

11.5.1.2. Établissement de communications sécurisées pour les clients de messagerie

L'utilisation du système de cryptage SSL pour les utilisateur d'IMAP et POP sur le serveur de messagerie est une opération relativement simple.

Créez tout d'abord un certificat SSL. Pour ce faire, il existe deux possibilités: vous pouvez faire la demande auprès d'une *Autorité de certification* (*AC*) pour un certificat SSL ou vous pouvez créer vous-même un certificat auto-signé.



Avertissement

Les certificats auto-signés ne devraient être utilisé qu'a des fins de test. Tout serveur utilisé dans un environnement de production devrait avoir recours à un certificat obtenu auprès d'une AC.

Pour créer un certificat SSL auto-signé pour IMAP, passez au répertoire `/usr/share/ssl/certs/` et tapez la commande suivante en étant connecté en tant que super-utilisateur:

```
make imapd.pem
```

Pour accomplir tout le processus, répondez à toutes les questions.

Afin de créer un certificat SSL auto-signé pour for POP, passez au répertoire `/usr/share/ssl/certs/` et tapez la commande suivante en étant connecté en tant que super-utilisateur:

```
make ipop3d.pem
```

Ici encore, répondez à toutes les questions pour accomplir tout le processus.

Une fois ces opérations terminées, utilisez la commande `/sbin/service` pour lancer le démon approprié (`imaps` ou `pop3s`). Configurez ensuite le service `imaps` ou le service `pop3s` afin leur démarrage s'effectue au niveau d'exécution approprié à l'aide d'un utilitaire initscript, comme l'**Outil**

de configuration des services (redhat-config-services). Reportez-vous à la Section 1.4.2 pour obtenir de plus amples informations sur les utilitaires initscript.

Il est également possible d'utiliser la commande stunnel en tant qu'enveloppeur de cryptage SSL placé autour des démons non-sécurisés standard imapd ou pop3d.

Le programme stunnel utilise des bibliothèques OpenSSL externes fournies avec Red Hat Linux, pour offrir un cryptage puissant et protéger les connexions. Il est recommandé de faire une demande de certificat SSL auprès d'une *Autorité de certification* (*AC*), mails il est également possible de créer un certificat auto-signé.

Pour créer un certificat SSL auto-signé, passez au répertoire /usr/share/ssl/certs/ et tapez la commande suivante:

```
make stunnel.pem
```

Ici encore, répondez à toutes les questions pour accomplir tout le processus.

Une fois le certificat créé, il est possible d'utiliser la commande stunnel pour démarrer le démon imapd à l'aide de la commande suivante:

```
/usr/sbin/stunnel -d 993 -l /usr/sbin/imapd imapd
```

Après l'exécution de cette commande, il est possible d'ouvrir un client de messagerie IMAP et d'établir une connexion au serveur de messagerie utilisant le système de cryptage SSL.

Pour lancer pop3d à l'aide de la commande stunnel, tapez la commande suivante:

```
/usr/sbin/stunnel -d 993 -l /usr/sbin/pop3d pop3d
```

Pour obtenir plus d'informations sur la façon d'utiliser stunnel, lisez la page de manuel relative à stunnel ou consultez les documents dans /usr/share/doc/stunnel-<*numéro-version*>/ directory.

11.6. Ressources supplémentaires

Ci-dessous figure une liste de la documentation supplémentaire relative aux applications de messagerie.

11.6.1. Documentation installée

- Les paquetages sendmail and sendmail-cf contiennent des informations sur la manière de configurer Sendmail.
  - /usr/share/doc/sendmail/README.cf — Informations sur m4, emplacements de fichier pour Sendmail, boîtes d'envoi prises en charge, façons d'accéder à des fonctions avancées, etc.
  - /usr/share/doc/sendmail/README — Informations sur la structure de répertoires de Sendmail, prise en charge de protocoles IDENT, détails sur les autorisations de répertoire et les problèmes communs que ces autorisations peuvent causer si elles ne sont pas configurées correctement.

  En outre, les pages de manuel relatives à sendmail et aliases contiennent des informations utiles sur les différentes options de Sendmail et la configuration adéquate du fichier Sendmail, /etc/mail/aliases.

- `/usr/share/doc/fetchmail-`<*numéro de version*> — Liste complète de fonctions Fetchmail dans le fichier `FEATURES` et document `FAQ` d'introduction.
- `/usr/share/doc/procmail-`<*numéro de version*> — Fichier `README` qui offre un aperçu de Procmail, fichier `FEATURES` qui explore toutes les fonctions du programme et fichier `FAQ` qui offre les réponses à de nombreuses questions fréquentes.

  Lorsque vous apprenez comment fonctionne Procmail et comment créer de nouvelles recettes, les pages de manuel suivantes sont précieuses:

  - `procmail` — offre un aperçu du fonctionnement de Procmail et des étapes de filtration du courrier.
  - `procmailrc` — explique le format de fichier `rc` utilisé pour créer des recettes.
  - `procmailex` — donne des exemples pratiques utiles de recettes Procmail.
  - `procmailsc` — explique la technique "weighted scoring" utilisée par Procmail pour vérifier s'il y a concordance entre une recette donnée et un message.
  - `/usr/share/doc/spamassassin-`<*numéro-version*>`/` — Ce répertoire contient de nombreuses informations sur SpamAssassin. Remplacez <*numéro-version*> par le numéro de version du paquetage `spamassassin`.

11.6.2. Sites Web utiles

- http://www.redhat.com/mirrors/LDP/HOWTO/Mail-Administrator-HOWTO.html — Fournit un aperçu du fonctionnement du courrier électronique et examine les solutions et configurations possibles de messagerie électronique, tant du côté serveur que client.
- http://www.redhat.com/mirrors/LDP/HOWTO/Mail-User-HOWTO/ — Examine le courrier électronique du point de vue de l'utilisateur, analyse diverses applications client de messagerie très utilisées et offre une introduction sur des sujets variés, tels que les alias, le réacheminement, la réponse automatique, les listes d'adresses, les filtres de courrier et les spams.
- http://www.redhat.com/mirrors/LDP/HOWTO/mini/Secure-POP+SSH.html — Explique une façon de récupérer du courrier POP en utilisant SSH avec le réacheminement de port, afin que les mots de passe et les messages soient transférés de manière sécurisée.
- http://www.sendmail.net/ — Contient des informations récentes, entrevues et articles relatifs à Sendmail, notamment un aperçu détaillé des nombreuses options disponibles.
- http://www.sendmail.org/ — Offre une explication technique très détaillée des fonctions de Sendmail et des exemples de configuration.
- http://tuxedo.org/~esr/fetchmail — Page d'accueil de Fetchmail, comprenant un manuel en ligne et une FAQ exhaustive.
- http://www.procmail.org/ — Page d'accueil de Procmail, avec des liens menant à diverses listes d'adresses de participants dédiées à Procmail, de même que de nombreux documents FAQ.
- http://www.ling.helsinki.fi/users/reriksso/procmail/mini-faq.html — Un excellent FAQ sur Procmail, offrant des conseils pour le résolution de problèmes, des informations au sujet du verrouillage de fichiers et l'utilisation de caractères génériques ('wildcards').
- http://www.uwasa.fi/~ts/info/proctips.html — Contient de nombreux conseils rendant l'utilisation de Procmail plus aisée. Ce site inclut des instructions sur la manière de tester les fichiers `.procmailrc` et d'utiliser le marquage de Procmail pour décider si une action donnée doit être exécutée ou non.
- http://www.spamassassin.org/ — Le site officiel du projet SpamAssassin.

11.6.3. Livres sur le sujet

- *Sendmail* de Bryan Costales avec Eric Allman et al; O'Reilly & Associates — Une bonne référence Sendmail, écrite avec l'aide du créateur original de Delivermail et Sendmail.
- *Removing the Spam: Email Processing and Filtering* de Geoff Mulligan; Addison-Wesley Publishing Company — Un livre examinant les diverses méthodes utilisées par les administrateurs de messagerie ayant recours à des outils établis, tels que Sendmail et Procmail, pour gérer les problèmes causés par les spams.
- *Internet Email Protocols: A Developer's Guide* de Kevin Johnson; Addison-Wesley Publishing Company — Fournit des informations détaillées sur les principaux protocoles de messagerie et la sécurité offerte par ceux-ci.
- *Managing IMAP* de Dianna Mullet et Kevin Mullet; O'Reilly & Associates — Explique les étapes nécessaires à la configuration d'un serveur IMAP.





Chapitre 12. Berkeley Internet Name Domain (BIND)

Sur la plupart des réseaux modernes, y compris l'Internet, les utilisateurs localisent les autres ordinateurs au moyen du nom. Ceci évite aux utilisateurs de devoir se rappeler de l'adresse réseau numérique des ressources réseau. La manière la plus efficace de configurer un réseau afin de permettre des connexions à base de nom consiste à établir un *Service de Nom de Domaine* (ou *DNS*, de l'anglais 'Domain Name Service') ou *serveur de noms* qui permet d'associer des noms d'hôte d'un réseau à des adresses numériques et vice-versa.

Le présent chapitre examine le serveur de noms inclus dans Red Hat Linux, *Berkeley Internet Name Domain* (*BIND*) serveur DNS, et met l'accent tout particulièrement sur la structure de ses fichiers de configuration et sur la manière de l'administrer aussi bien localement qu'à distance.

Pour obtenir des instructions sur la configuration de BIND à l'aide de l'application graphique **Outil de configuration Bind** (`redhat-config-bind`), reportez-vous au chapitre intitulé *Configuration de BIND* du *Guide de personnalisation de Red Hat Linux*.



Avertissement

Si vous utilisez l'**Outil de configuration Bind**, ne modifiez manuellement aucun des fichiers de configuration BIND car tout changement sera écrasé lors d'une utilisation postérieure de l'application **Outil de configuration Bind**.

12.1. Introduction au DNS

Lorsque les hôtes d'un réseau se connectent entre eux au moyen d'un nom d'hôte, auquel on se réfère également sous le terme *nom de domaine pleinement qualifié ou 'fully qualified domain name' (FQDN)*, le DNS est utilisé pour associer les noms des différents ordinateurs à l'adresse IP de l'hôte.

L'utilisation du DNS et du FQDN offre aux administrateurs système de nombreux avantages et leur permet, en outre, de changer facilement l'adresse IP d'un hôte sans avoir d'impact sur les requêtes basées sur le nom envoyées à cet ordinateur. Inversement, les administrateurs peuvent décider des machines qui traiteront une requête basée sur le nom.

Le service DNS est normalement mis en oeuvre grâce à des serveurs centralisés qui font autorité pour certains domaines, et se réfèrent à d'autres serveurs DNS pour d'autres domaines.

Lorsqu'un hôte client demande des informations au serveur de noms, il se connecte généralement sur le port 53. Le serveur de noms tente alors de résoudre le FQDN d'après sa bibliothèque de solutions qui peut contenir des informations importantes sur l'hôte demandé ou des données mise en cache suite à une requête antérieure. Si le serveur de noms ne possède pas encore la réponse dans sa bibliothèque de solutions, il se tourne vers d'autres serveurs de noms, appelés *serveurs de noms root* (ou serveurs de noms racines), afin de déterminer les serveurs de noms faisant autorité pour le FQDN en question. Grâce à ces informations, il effectuera ensuite une requête auprès des serveurs de noms faisant autorité pour déterminer l'adresse IP de l'hôte en question. S'il effectue une opération dans le sens inverse (reverse lookup), c'est la même procédure qui est utilisée, si ce n'est que la requête est présentée avec une adresse IP inconnue au lieu d'un nom.

12.1.1. Zones de serveurs de noms

Sur Internet, le FQDN d'un hôte peut être structuré en sections qui sont ensuite organisées hiérarchiquement, comme un arbre avec un tronc principal, des branches primaires, des branches secondaires, etc. Prenons, par exemple, le FQDN suivant:

```
bob.sales.example.com
```

Lorsque vous regardez un FQDN pour trouver l'adresse IP qui renvoie à un système particulier, lisez le nom de droite à gauche et chaque niveau de la hiérarchie divisé par des points (`.`). Dans notre exemple, le `com` définit le *domaine de niveau supérieur* pour ce FQDN. Le nom `example` est un sous-domaine de `com` alors que `sales` est un sous-domaine de `example`. Le nom le plus à gauche `bob`, identifie une machine particulière. machine.

À l'exception du nom de domaine, chaque section s'appelle une *zone* et définit un *espace de nom* particulier. Un espace de nom contrôle l'attribution des noms des sous-domaines à sa gauche. Alors que cet exemple ne contient que deux sous-domaines, un FQDN doit contenir au moins un sous-domaine mais peut en inclure beaucoup plus, selon l'organisation de l'espace de nom choisie.

Les zones sont définies sur des serveurs de noms qui font autorité par l'intermédiaire; *fichiers de zone*, décrivant entre autres, l'espace de nom de cette zone, les serveurs de courrier qui doivent être utilisés pour un domaine ou sous-domaine particulier. Les fichiers de zone sont stockés sur des *serveurs de noms primaires* (aussi appelés *serveurs de noms maîtres*), qui font vraiment autorité et sont l'endroit où des changements peuvent être apportés aux fichiers; les *serveurs de noms primaires secondaires* (ou *serveurs de noms esclaves*) quant à eux reçoivent leurs fichiers de zone des serveurs de noms primaires. Tout serveur de noms peut être simultanément maître ou esclave pour différentes zones et peut aussi être considéré comme faisant autorité pour de multiples zones. Tout cela dépend de la configuration du serveur de noms.

12.1.2. Types de serveurs de noms

Il existe quatre types de configuration de serveurs de noms:

- *maître* — Stocke les enregistrements de zone originaux faisant autorité pour un certain espace de nom et répond aux questions d'autres serveurs de noms qui cherchent des réponses concernant cet espace de nom.
- *esclave* — Répond aux requêtes d'autres serveurs de noms concernant les espaces de nom pour lesquels il est considéré comme faisant autorité. Les serveurs de noms esclaves reçoivent leurs informations d'espace de noms des serveurs de noms maîtres.
- *caching-only* — Offre des services de résolution nom vers IP mais ne fait pas autorité dans n'importe quelle zone. Les réponses pour toutes les résolutions sont placées en cache dans une base de données stockée en mémoire pour une période établie qui est spécifiée par l'enregistrement de zone importé.
- *retransmission* — Fait suivre des requêtes pour résolution à une liste spécifique de serveurs de noms. Si aucun des serveurs de noms spécifiés ne peut effectuer la résolution, le processus s'arrête et la résolution a échoué.

Un serveur de noms peut être d'un ou plusieurs de ces types. Par exemple, un serveur de noms peut être non seulement maître pour certaines zones, esclave pour d'autres mais peut également offrir seulement la transmission d'une résolution pour d'autres encore.

12.1.3. BIND en tant que serveur de noms

Le serveur de noms BIND fournit ses services de résolution de noms à l'aide du démon `/usr/sbin/named`. BIND contient également un utilitaire d'administration appelé `/usr/sbin/rndc`. De plus amples informations sur `rndc` sont disponibles dans la Section 12.4.

BIND stocke ses fichiers de configuration dans les deux endroits suivants:

- le fichier `/etc/named.conf` — le fichier de configuration du démon `named`.
- le répertoire `/var/named/` — le répertoire de travail de `named` qui stocke les fichiers de zone, de statistiques et les fichiers de cache.

Les sections suivantes examinent les fichiers de configuration de manières plus détaillée.

12.2. `/etc/named.conf`

Le fichier `named.conf` est une suite de déclarations utilisant des options insérées qui sont placées entre accolades, `{ }`. Les administrateurs doivent être très prudents lorsqu'ils modifient le fichier `named.conf` et doivent veillez tout particulièrement à ne pas faire de fautes de syntaxe car des erreurs mineures en apparence empêcheront le démarrage du service `named`.

**Avertissement**

> *Ne modifiez pas* manuellement le fichier `/etc/named.conf` ou tout autre fichier dans le répertoire `/var/named/` si vous n'utilisez pas l'**Outil de configuration Bind** au moment des modifications. Tout changement manuel dans ce fichier ou dans tout fichier de ce répertoire sera écrasé lors de la prochaine utilisation de l'**Outil de configuration Bind**.

Un fichier `named.conf` typique est organisé de manière semblable à l'extrait ci-dessous:

```
<déclaration-1> ["<déclaration-1-nom>"] [<déclaration-1-classe>] {
   <option-1>;
   <option-2>;
   <option-N>;
};

<déclaration-2> ["<déclaration-2-nom>"] [<déclaration-2-classe>] {
   <option-1>;
   <option-2>;
   <option-N>;
};

<déclaration-N> ["<déclaration-N-nom>"] [<statement-N-classe>] {
   <option-1>;
   <option-2>;
   <option-N>;
};
```

12.2.1. Types de déclarations courants

Les types de déclarations suivants sont couramment utilisés dans `/etc/named.conf`:

12.2.1.1. Déclaration acl

La déclaration `acl` (ou déclaration de contrôle d'accès) définit des groupes d'hôtes qui peuvent ensuite être autorisés ou non à accéder au serveur de noms.

Une déclaration `acl` se présente sous le format suivant:

```
acl <acl-nom> {
    <élément-correspondant>;
    [<élément-correspondant>; ...]
};
```

Dans cette déclaration, remplacez <*acl-nom*> par le nom de la liste du contrôle d'accès et remplacez <*élément-correspondant*> en séparant les adresses IP par un point virgule. La plupart du temps, une adresse IP individuelle ou la notation réseau de l'IP (comme par exemple, `10.0.1.0/24`) est utilisée pour identifier les adresses IP dans la déclaration `acl`.

Les listes de contrôle d'accès suivantes sont déjà définies en tant que mots-clés afin de simplifier la configuration:

- `any` — correspond à toutes les adresses IP.
- `localhost` — correspond à toute adresse IP utilisée par le système local.
- `localnets` — correspond à toute adresse IP sur tout réseau auquel le système local est connecté.
- `none` — ne correspond à aucune adresse IP.

Lorsqu'elles sont utilisées avec d'autres déclarations (comme par exemple, la déclaration `options`), les déclarations `acl` peuvent se révéler très utiles pour éviter la mauvaise utilisation d'un serveur de noms BIND.

L'exemple ci-dessous établit deux listes de contrôle d'accès et utilise une déclaration `options` pour définir la manière dont elles seront traitées par le serveur de nom:

```
  acl black-hats {
    10.0.2.0/24;
    192.168.0.0/24;
 };

 acl red-hats {
    10.0.1.0/24;
 };

 options {
    blackhole { black-hats; };
    allow-query { red-hats; };
    allow-recursion { red-hats; };
 }
```

Cet exemple comporte deux listes de contôle d'accès, `black-hats` et `red-hats`. Les hôtes de la liste `black-hats` se voient dénier l'accès au serveur de noms, alors que ceux de la liste `red-hats` se voient donner un accès normal.

12.2.1.2. Déclaration `include`

La déclaration `include` permet à des fichiers de faire partie d'un fichier `named.conf`. Ce faisant, des données de configurations critiques (comme `keys`, par exemple) dans un fichier séparé doté de permissions restreintes.

Une déclaration `include` se présente sous le format suivant:

```
include  "<nom-fichier>"
```

Dans cette déclaration, `<nom-fichier>` est remplacé par le chemin d'accès absolu vers un fichier.

12.2.1.3. Déclaration `options`

La déclaration `options` définit les options globales de configuration de serveur et établit des valeurs par défaut pour les autres déclarations. Cette déclaration peut être utilisée en autres pour spécifier l'emplacement du répertoire de travail `named`, ou pour déterminer les types de requêtes autorisés.

La déclaration `options` se présente sous le format suivant:

```
options {
        <option>;
 [<option>; ...]
};
```

Dans cette déclaration, les directives `<option>` sont remplacées par une option valide.

Ci-dessous figure une liste des options couramment utilisées:

- `allow-query` — spécifie les hôtes autorisé à interroger ce serveur de noms. Par défaut, tous les hôtes sont autorisés à interroger le serveur de noms. Une liste de contrôle d'accès ou un ensemble d'adresses IP ou de réseaux peuvent être utilisés ici afin de n'autoriser que des hôtes précis à interroger le serveur de noms.
- `allow-recursion` — semblable à `allow-query`, cette option s'applique à des demandes récursives. Par défaut, tous les hôtes sont autorisés à effectuer des demandes récursives sur le serveur de noms.
- `blackhole` — spécifie les hôtes qui ne sont pas autorisés à interroger le serveur de noms.
- `directory` — change le répertoire de travail `named` (`/var/named/`) pour une valeur autre que `/var/named/`, la valeur par défaut.
- `forward` — contrôle le comportement de retransmission d'une directive `forwarders`.

  Les options suivantes sont acceptées:

  - `first` — établit que les serveurs de noms spécifiés dans la directive `forwarders` soient interrogés avant que `named` ne tente de résoudre le nom lui-même.
  - `only` — spécifie que `named` ne doit pas tenter d'effectuer lui-même une résolution de nom dans le cas où des demandes vers les serveurs de noms spécifiés dans la directive `forwarders` échoueraient.

- `forwarders` — spécifie une liste d'adresses IP valides correspondant aux serveurs de noms vers lesquels les requêtes devraient être envoyées pour la résolution.
- `listen-on` — spécifie l'interface réseau sur laquelle `named` prend note des requêtes. Par défaut, toutes les interfaces sont utilisées.

  De cette manière, si le serveur DNS sert également de portail, BIND peut être configuré de telle sorte qu'il ne réponde qu'aux requêtes en provenance de l'un des réseaux.

Une directive `listen-on` peut ressembler à l'extrait ci-dessous:

```
options {
   listen-on { 10.0.1.1; };
};
```

De cette manière, seules les requêtes qui proviennent de l'interface de réseau servant le réseau privé (`10.0.1.1`) seront acceptées.

- `notify` — détermine si `named` envoie une notification aux serveurs esclaves quand une zone est mise à jour. Il accepte les options suivantes:
  - `yes` — notifie les serveurs esclaves.
  - `no` — ne notifie les serveurs esclaves.
  - `explicit` — notifie seulement les serveurs esclaves spécifiés dans une liste `also-notify` à l'intérieur d'une déclaration de zone.

- `pid-file` — spécifie l'emplacement du fichier de processus ID créé par `named`.
- `statistics-file` — spécifie un autre emplacement des fichiers de statistiques. Par défaut, les `named` sont enregistrées dans le fichier `/var/named/named.stats`.

De nombreuses autres options sont également disponibles, dont beaucoup dépendant l'une de l'autre pour fonctionner correctement. Consultez le document *BIND 9 Administrator Reference Manual* dans la Section 12.7.1 et la page de manuel relative à `bind.conf` pour de plus amples informations.

12.2.1.4. Déclaration de `zone`

Une déclaration de `zone` définit les caractéristiques d'une zone tels que l'emplacement de ses fichiers de configuration et les options spécifiques à la zone. Cette déclaration peutêtre utilisée pour remplacer les déclarations globales d'`options` statements.

Une déclaration de `zone` se présente sous le format suivant:

```
zone <zone-nom> <zone-classe> {
     <zone-options>;
     [<zone-options>; ...]
};
```

Dans la déclaration, *<zone-nom>* correspond au nom de la zone, *<zone-classe>* à la classe optionnelle de la zone et *<zone-options>* représente une liste des options caractérisant la zone.

L'attribut *<zone-nom>* de la déclaration de zone est particulièrement important, puisqu'il représente la valeur par défaut assignée à la directive `$ORIGIN` utilisés au sein du fichier de zone correspondant qui se trouve dans le répertoire `/var/named/`. Le démon `named` attache le nom de la zone à tout nom de domaine qui n'est pas pleinement qualifié, listé dans le fichier de zone.

Par exemple, si une déclaration de `zone` définit l'espace de nom pour `example.com`, utilisez `example.com` comme *<zone-nom>* afin qu'il soit placé à la fin des noms d'hôtes au sein du fichier de zone `example.com`.

Pour de plus amples informations sur les fichiers de zone, reportez-vous à la Section 12.3.

Parmi les options les plus courantes de la déclaration de `zone` figurent:

- `allow-query` — spécifie les clients qui sont autorisés à requérir des informations à propos de cette zone. Par défaut toutes les requêtes d'informations sont autorisées.
- `allow-transfer` — spécifie les serveurs esclaves qui sont autorisés à requérir un transfert des informations de la zone. Par défaut toutes les requêtes de transfert sont autorisées.

- `allow-update` — spécifie les hôtes qui sont autorisés à mettre à jour dynamiquement des informations dans leur zone. Par défaut aucune requête de mise à jour dynamique n'est autorisée.

  Soyez très prudent lorsque vous autorisez des hôtes à mettre à jour des informations à propos de leur zone. Ne mettez en oeuvre cette option que si vous accordez une confiance absolue à l'hôte. De manière générale, il est préférable de laisser un administrateur mettre à jour manuellement les enregistrements de la zone et recharger le service `named` service.
- `file` — spécifie le nom du fichier qui contient les données de configuration de la zone, dans le répertoire de travail `named`.
- `masters` — l'option `masters` établit une liste des adresses IP à partir desquelles demander des informations sur la zone faisant autorité. Cette option ne doit être utilisée que si la zone est définie comme de `type slave`.
- `notify` — établit si `named` notifie les serveurs esclaves lorsqu'une zone est mise à jour. Les options suivantes sont acceptées:
  - `yes` — notifie les serveurs esclaves.
  - `no` — ne notifie pas les serveurs esclaves.
  - `explicit` — notifie seulement les serveurs esclaves spécifiés dans une liste `also-notify` à l'intérieur d'une déclaration de zone.
- `type` — définit le type de zone. Les types énumérés ci-dessous peuvent être utilisés.

  Ci-après figure une liste des options valides:
  - `forward` — retransmet toutes les requêtes d'informations à propos de cette zone vers d'autres serveurs de noms
  - `hint` — un type spécial de zone utilisé pour diriger des transactions vers les serveurs de noms racines qui résolvent des requêtes lorsqu'une zone n'est pas connue autrement. Aucune configuration au-delà de la valeur par défaut n'est nécessaire avec une zone `hint`.
  - `master` — désigne le serveur de noms faisant autorité pour cette zone. Une zone devrait être configurée comme de type `master` (maître) si les fichiers de configuration de la zone se trouvent sur le système.
  - `slave` — désigne le serveur de noms comme serveur esclave pour cette zone. Cette option spécifie également l'adresse IP du serveur de noms maître pour cette zone.
- `zone-statistics` — configure `named` pour qu'il conserve des statistiques concernant cette zone, en les écrivant soit dans l'emplacement par défaut (`/var/named/named.stats`) soit à l'emplacement expressément désigné par l'option `statistics-file` dans la déclaration `server`. Reportez-vous à la Section 12.2.2 pour de plus amples informations sur la déclaration `server`.

12.2.1.5. Exemples de déclarations `zone` Statements

La plupart des changements apportés au fichier `/etc/named.conf` d'un serveur de noms maître ou esclave implique l'ajout, la modification ou la suppression de déclarations de `zone`. Alors que ces déclarations de `zone` peuvent contenir de nombreuses options, la plupart des noms de serveurs n'en ont besoin que de peu pour fonctionner de manière efficace. Les déclarations de `zone` suivantes sont des exemples très élémentaires illustrant une relation de serveurs de noms maître/esclave.

Ci-dessous se trouve un exemple de déclaration de `zone` pour le serveur de noms primaire hébergeant `example.com` (`192.168.0.1`):

```
zone "example.com" IN {
```

```
  type master;
  file "example.com.zone";
  allow-update { none; };
};
```

Dans cette déclaration, la zone est identifiée en tant que `example.com`, le type est défini comme `master` et le service `named` a comme instruction de lire le fichier `/var/named/example.com.zone`. Elles indique à `named` de refuser la mise à jour à tout autre hôte.

La déclaration de `zone` d'un serveur esclave pour `example.com` est légèrement différente de l'exemple précédent. Pour un serveur esclave, le type retenu est `slave` et une directive indiquant à `named` l'adresse IP du serveur maître remplace la ligne `allow-update`.

La déclaration de `zone` d'un serveur esclave pour example.com pourrait ressembler à l'extrait ci-dessous:

```
zone "example.com" {
  type slave;
  file "example.com.zone";
  masters { 192.168.0.1; };
};
```

Cette déclaration de `zone` configure `named` sur le serveur esclave de manière à ce qu'il cherche le serveur maître à l'adresse IP `192.168.0.1` pour y trouver les informations sur la zone appelée `example.com`. Les informations que le serveur esclave reçoit du serveur maître sont enregistrées dans le fichier `/var/named/example.com.zone`.

12.2.2. Autres types de déclarations

Ci-dessous se trouve une liste de types de déclarations disponibles au sein de `named.conf` mais utilisés moins fréquemment.

- `controls` — configure diverses contraintes de sécurité nécessaires â l'utilisation de la commande `rndc` pour administrer le service `named`.

  Consultez la Section 12.4.1 pour voir ce à quoi devrait ressembler la déclaration `controls`, y compris les options diverses qui ne peuvent être utilisées qu'avec elle.

- `key "<nom-clé>"` — définit une clé spécifique par nom. Les clés servent à valider diverses actions, comme les mises à jour sécurisées ou l'utilisation de la commande `rndc`. Deux options sont utilisées avec `key`:

  - `algorithm <nom-algorithme>` — le type d'algorithme utilisé, comme par exemple `dsa` ou `hmac-md5`.
  - `secret "<valeur-clé>"` — La clé cryptée.

  Reportez-vous à la Section 12.4.2 pour obtenir des instructions sur l'écriture d'une déclaration `key`.

- `logging` — permet d'utiliser de multiples types de logs (ou journaux), appelés des *channels*. En utilisant l'option `channel` dans la déclaration `logging`, il est possible de construire un type de journal personnalisé, avec son propre nom de fichier (`file`), sa limite de taille (`size`), sa version (`version`) et son niveau d'importance (`severity`). Une fois qu'un channel personnalisé a été défini, une option `category` est utilisée pour catégoriser le channel et commencer le logging quand `named` est redémarrée.

  Par défaut, `named` envoie des messages de log standards au démon `syslog`, qui les place dans `/var/log/messages`. Ceci se produit car plusieurs canaux standards sont compris dans BIND, avec plusieurs niveaux d'importance, comme celui qui traite les messages de logging (ou journalisation) informationnels (`default_syslog`) et celui qui traite spécifiquement les messages de

débogage (default_debug). Une catégorie par défaut, appelée default, utilise les canaux compris dans BIND pour accomplir la journalisation normale, sans configuration spéciale.

La personnalisation du processus de journalisation (logging) peut être un processus très détaillé qui dépasse le cadre du présent chapitre. Pour obtenir plus d'informations sur la création de logs personnalisés dans BIND, consultez le *BIND 9 Administrator Reference Manual* dans la Section 12.7.1.

- server — définit des options particulières qui affectent la façon dont named doit se comporter envers les serveurs de noms distants, particulièrement en ce qui concerne les notifications et les transferts de zone.

  L'option transfer-format détermine si un enregistrement de ressource est envoyé avec chaque message (one-answer) ou si des enregistrements de ressource multiples sont envoyés avec chaque message (many-answers). Alors que many-answers est plus efficace, seuls les plus récents serveurs de noms BIND peuvent la comprendre.

- trusted-keys — contient des clés publiques assorties utilisées pour un DNS sécurisé(DNSSEC). Consultez la Section 12.5.3 pour de plus amples informations sur la sécurité sous BIND.

- view "<*nom-vue*>" — crée des vues spéciales selon l'hôte qui contacte le serveur de noms. Ceci permet à certains hôtes de recevoir une réponse concernant une zone particulière alors que d'autres hôtes reçoivent des informations totalement différentes. Certains hôtes de confiance peuvent également se voir accorder l'accès à certaines zones alors que d'autres hôtes qui ne sont des dignes de confiance doivent limiter leurs requêtes à d'autres zones.

  Vous pouvez utiliser de multiples vues, pour autant que leurs noms soient uniques. L'option match-clients spécifie les adresses IP qui s'appliquent à une vue particulière. Toute déclaration options peut aussi être utilisée dans une vue, avec priorité sur les options globales déjà configurées pour named. La plupart des déclarations view contiennent de multiples déclarations zone qui s'appliquent à la liste match-clients. L'ordre dans lequel les déclarations view sont listées est important, puisque c'est la première déclaration view qui correspond à l'adresse IP d'un client, qui est utilisée.

  Consultez la Section 12.5.2 pour obtenir plus d'informations sur la déclaration view.

12.2.3. Balises de commentaire

La liste suivante regroupe les balises (ou tags) de commentaire valides utilisés dans named.conf:

- // — lorsque ce symbole est placé en début de ligne, cette dernière n'est pas prise en compte par named.
- # — lorsque ce symbole est placé en début de ligne, cette dernière n'est pas prise en compte par named.
- /* et */ — lorsque du texte est placé entre ces symboles, le bloc de texte en question n'est pas prise en compte par named.

12.3. Fichiers de zone

Les *Fichiers de zone* contiennent des informations sur un espace de nom particulier et sont stockés dans le répertoire de travail named qui est par défaut /var/named/. Chaque fichier de zone est nommé selon les données d'options de file dans la déclaration zone, et ce, généralement d'une manière qui se réfère au domaine en question et identifie le fichier comme contenant des données de zone, telles que example.com.zone.

Chaque fichier de zone peut contenir des directives et enregistrements de ressources. Les *directives* donnent au serveur de noms l'instruction d'effectuer une certaine tâche ou d'appliquer des paramètres spéciaux à la zone. Les *enregistrements de ressources* définissent les paramètres de la zone, assignant des identités aux hôtes individuels. Les directives sont facultatives, mais les enregistrements de ressources sont requis pour fournir un service de noms à une zone.

Toutes les directives et enregistrements de ressources doivent se situer sur leur propre ligne.

Des commentaires peuvent être placés dans les fichiers de zone après les caractères points-virgules (`;`).

12.3.1. Directives de fichiers de zone

Les directives sont identifiées par le symbole dollar (`$`) suivit du nom de la directive. Elles apparaissent généralement en haut du fichier de zone.

Les directives les plus couramment utilisées sont les suivantes:

- `$INCLUDE` — configure `named` de façon à ce qu'il inclue un autre fichier de zone dans ce fichier de zone à l'endroit où la directive apparaît. Cela permet de stocker des configurations de zone supplémentaires à l'écart du fichier de zone principal.
- `$ORIGIN` — attache le nom de domaine à tout enregistrement non-qualifié, comme ceux qui spécifient seulement l'hôte et rien de plus.

  Un fichier de zone peut par exemple, contenir la ligne suivante:
  ```
  $ORIGIN example.com
  ```
  Tout nom utilisé dans les enregistrements de ressources et ne finissant pas par un point (`.`) se verront ajouter le nom de domaine `example.com`.

  Remarque

  L'utilisation de la directive `$ORIGIN` n'est pas nécessaire si l'on nomme la zone dans `/etc/named.conf` parce que le nom de la zone est utilisé par défaut,comme la valeur de la directive `$ORIGIN`

- `$TTL` — règle la valeur par défaut de *Time to Live (TTL)* (ou temps de vie) pour la zone. Cette valeur exprimée en secondes, correspond à la durée pendant laquelle les enregistrements de ressources de la zone resteront valides. Chaque enregistrement de ressources peut contenir sa propre valeur TTL, qui remplace alors cette directive.

  En accroissant cette valeur, les serveurs de noms distants peuvent mettre en cache ces informations de zone pendant plus longtemps. Cela réduit le nombre de requêtes effectuées au sujet de cette zone, mais rallonge également le temps nécessaire pour la prolifération des changements des enregistrements de ressources.

12.3.2. Enregistrements de ressources de fichiers de zone

Les enregistrements de ressources représentent le premier composant d'un fichier de zone.

Il existe de nombreux types différents d'enregistrements de ressources de fichiers de zone. Ceux énumérés ci-dessous sont néanmoins les plus fréquemment utilisés:

- `A` — enregistrement d'adresse qui spécifie une adresse IP à assigner à un nom, comme dans l'exemple ci-dessous:
  ```
  <hôte>      IN     A      <adresse-IP>
  ```

Si la valeur <*hôte*> est omise, alors un enregistrement `A` renvoie à une adresse IP par défaut pour le haut de l'espace de nom. Ce système est la cible de toutes les requêtes non-FQDN.

Examinons les exemples d'enregistrement `A` suivants pour le fichier de zone `example.com`:

```
             IN     A       10.0.1.3
server1      IN     A       10.0.1.5
```

Les requêtes pour `example.com` sont orientées vers 10.0.1.3, alors que les requêtes pour `server1.example.com` sont orientées vers 10.0.1.5.

- `CNAME` — enregistrement de nom canonique mappant un nom à un autre. Ce type d'enregistrement est pus connu sous le nom d'enregistrement d'alias.

  L'exemple suivant donne à `named` l'instruction d'envoyer toute requête au <*nom-alias*> qui sera sera alors orientée vers l'hôte, <*nom-réel*>. Les enregistrements `CNAME` sont généralement utilisés pour orienter vers les services qui utilisent un procédé commun de nommage, comme par exemple, `www` pour les serveurs Web.

  ```
  <nom-alias>     IN     CNAME      <nom-réel>
  ```

  Dans l'exemple suivant, un enregistrement `A` fixe un nom d'hôte à une adresse IP alors qu'un enregistrement `CNAME` y oriente le nom d'hôte `www` le fréquemment utilisé.

  ```
  server1      IN     A       10.0.1.5
  www          IN     CNAME   server1
  ```

- `MX` — enregistrement Mail eXchange, qui indique où doit se diriger le courrier envoyé à un nom d'espace particulier contrôlé par cette zone.

  ```
        IN     MX     <valeur-préférence>   <nom-serveur-email>
  ```

  Dans cet exemple, <*valeur-préférence*> permet de classer numériquement les serveurs de mail pour un espace de nom, en donnant une préférence à certains systèmes de courrier sur d'autres. L'enregistrement de ressource `MX` doté de la <*valeur-préférence*> la plus basse est préféré aux autres. Toutefois, de multiples serveurs de courrier peuvent avoir la même valeur pour distribuer de manière égale le trafic des emails entre eux.

  L'option <*nom-serveur-email*> peut être un nom d'hôte ou un FQDN.

  ```
        IN     MX     10     mail.example.com.
        IN     MX     20     mail2.example.com.
  ```

  Dans cet exemple, le premier serveur de courrier `mail.example.com` est préféré au serveur de courrier `mail2.example.com` lors de la réception des emails destinés au domaine `example.com`.

- `NS` — enregistrement de serveur de noms (NameServer) annonçant les serveurs de noms faisant autorité pour une zone particulière.

  Ci-dessous figure un exemple d'enregistrement `NS`:

  ```
        IN     NS     <nom-serveur de noms>
  ```

  L'option <*nom-serveur de noms*> devrait correspondre à un FQDN.

  Ensuite, deux serveurs de noms sont répertoriés comme faisant autorité pour le domaine. Le fait que ces serveurs de noms soient esclaves ou que l'un soit maître n'a pas d'importance; ils sont tous les deux considérés comme faisant autorisé.

  ```
        IN     NS     dns1.example.com.
        IN     NS     dns2.example.com.
  ```

- `PTR` — enregistrement PoinTeR, conçu pour orienter vers une autre partie de l'espace de nom.

  Les enregistrements `PTR` servent essentiellement à la résolution inverse des noms, puisqu'ils réorientent les adresses IP vers un nom particulier. Consultez la Section 12.3.4 pour obtenir des exemples supplémentaires d'utilisations d'enregistrements `PTR`.

- `SOA` — enregistrement "Start Of Authority", proclamant des informations importantes faisant autorité à propos d'un espace de nom pour le serveur de noms.

  Situé après les directives, un enregistrement de ressources `SOA` est le premier enregistrement de ressources dans un fichier de zone.

L'exemple qui suit indique la structure de base d'un enregistrement `SOA`:

```
@     IN     SOA    <serveur-noms-primaire>     <hostmaster-email> (
                    <numéro-série>
                    <temps-actualisation>
                    <temps-nouvel essai>
                    <temps-expiration>
                    <TTL-minimum> )
```

Le symbole `@` place la directive `$ORIGIN` (ou le nom de zone, si la directive `$ORIGIN` n'est pas installée) en tant qu'espace de nom défini par le présent enregistrement de ressources `SOA`. Le serveur de noms primaire faisant autorité pour ce domaine est utilisé pour le <*serveurs-noms-primaire*> et l'adresse email de la personne à contacter à propos de cet espace de nom est remplacée par <*email-hôte maître*>.

The <*numéro-série*> est incrémentée chaque fois que vous changez le fichier de zone afin que `named` sache qu'il doit recharger cette zone. La valeur <*temps-actualisation*> indique à tout serveur esclave combien de temps il doit attendre avant de demander au serveur de noms maître si des changements ont été effectués dans la zone. La valeur <*numéro-série*> est utilisée par le serveur esclave pour déterminer s'il est en train d'utiliser des données de zone périmées et doit donc les rafraîchir.

La valeur <*temps-nouvel essai*> précise au serveur de noms esclave l'intervalle pendant lequel il doit attendre avant d'émettre une autre requête de rafraîchissement, au cas où le serveur de noms maître ne répondrait pas. Si le serveur maître n'a pas répondu à une requête de rafraîchissement avant que la durée indiquée dans <*temps-expiration*> ne se soit écoulée, le serveur esclave cesse de répondre en tant qu'autorité pour les requêtes au sujet de cet espace de nom.

La valeur <*TTL-minimum*> demande que d'autres serveurs de noms placent en cache les informations pour cette zone pendant au moins cette durée (en secondes).

Dans BIND, tous les durées sont exprimées en secondes. Toutefois, vous pouvez aussi utiliser des abréviations pour des unités de temps autres que des secondes, comme les minutes (`M`), heures (`H`), jours (`D`) et semaines (`W`). Le tableau de la Tableau 12-1 montre une durée en secondes et la période équivalente dans un autre format.

Secondes	Autres unités de temps
60	1M
1800	30M
3600	1H
10800	3H
21600	6H
43200	12H
86400	1D
259200	3D
604800	1W
31536000	365D

Tableau 12-1. Les secondes comparées à d'autres unités de temps

L'exemple suivant montre ce à quoi l'enregistrement d'une ressource de base `SOA` peut ressembler lorsqu'il est configuré avec des valeurs réelles.

```
@     IN     SOA    dns1.example.com.     hostmaster.example.com. (
                    2001062501 ; serial
                    21600      ; refresh after 6 hours
```

```
                    3600        ; retry after 1 hour
                    604800      ; expire after 1 week
                    86400 )     ; minimum TTL of 1 day
```

12.3.3. Exemples de fichiers de zone

Si on les observe individuellement, les directives et enregistrements de ressources peuvent être difficiles à comprendre. Cependant, tout devient beaucoup plus simple lorsqu'on peut les observer ensemble dans un seul fichier commun.

L'exemple suivant illustre un fichier de zone très élémentaire.

```
$ORIGIN example.com
$TTL 86400
@      IN     SOA    dns1.example.com.     hostmaster.example.com. (
                     2001062501 ; serial
                     21600      ; refresh after 6 hours
                     3600       ; retry after 1 hour
                     604800     ; expire after 1 week
                     86400 )    ; minimum TTL of 1 day

       IN     NS     dns1.example.com.
       IN     NS     dns2.example.com.

       IN     MX     10     mail.example.com.
       IN     MX     20     mail2.example.com.

              IN     A       10.0.1.5

server1       IN     A       10.0.1.5
server2       IN     A       10.0.1.7
dns1          IN     A       10.0.1.2
dns2          IN     A       10.0.1.3

ftp           IN     CNAME   server1
mail          IN     CNAME   server1
mail2         IN     CNAME   server2
www           IN     CNAME   server2
```

Dans cet exemple sont utilisées des directives et des valeurs `SOA` standard. Les serveurs de noms faisant autorité seront `dns1.example.com` et `dns2.example.com`, qui ont des enregistrements `A` les liant respectivement à `10.0.1.2` et `10.0.1.3`.

Les serveurs de courrier configurés par les enregistrements `MX` orientent vers les serveurs `server1` et `server2` au moyen des enregistrements `CNAME` Puisque les noms des serveurs `server1` et `server2` ne finissent pas par un point (`.`), le domaine `$ORIGIN` est attaché, rallongeant le nom en `server1.example.com` et `server2.example.com`. Grâce aux enregistrements de ressources `A` associés, leurs adresses IP peuvent être déterminées.

Les services FTP et Web services, disponibles aux noms standard `ftp.example.com` et `www.example.com`, sont orientés vers les serveurs appropriés en utilisant les enregistrements `CNAME`.

12.3.4. Fichiers de résolution de noms inversée

Un fichier de résolution de nom inversée sert à traduire une adresse IP dans un espace de nom particulier en un FQDN. Il ressemble beaucoup à un fichier de zone standard, si ce n'est que les enregistrements de ressources `PTR` servent à lier les adresses IP au nom d'un domaine pleinement qualifié.

Un enregistrement `PTR` ressemble à ce qui suit:

```
<dernier-chiffre-IP>       IN     PTR    <FQDN-du-système>
```

Le <*dernier-chiffre-IP*> fait référence au dernier chiffre dans une adresse IP qui doit orienter vers le FQDN d'un système particulier.

Dans l'exemple suivant, les adresses IP allant de `10.0.1.20` à `10.0.1.25` orientent vers les FQDN correspondants.

```
$ORIGIN 1.0.10.in-addr.arpa
$TTL 86400
@      IN     SOA    dns1.example.com.     hostmaster.example.com. (
                     2001062501 ; serial
                     21600      ; refresh after 6 hours
                     3600       ; retry after 1 hour
                     604800     ; expire after 1 week
                     86400 )    ; minimum TTL of 1 day

       IN     NS     dns1.example.com.
       IN     NS     dns2.example.com.

20     IN     PTR    alice.example.com.
21     IN     PTR    betty.example.com.
22     IN     PTR    charlie.example.com.
23     IN     PTR    doug.example.com.
24     IN     PTR    ernest.example.com.
25     IN     PTR    fanny.example.com.
```

Ce fichier de zone serait mis en service avec une déclaration `zone` dans le fichier `named.conf` similaire à l'extrait qui suit:

```
zone "1.0.10.in-addr.arpa" IN {
  type master;
  file "example.com.rr.zone";
  allow-update { none; };
};
```

Il existe peu de différences entre cet exemple et une déclaration `zone` standard, si ce n'est dans la manière de nommer l'hôte. Notez qu'une zone de résolution de noms inversée nécessite que les trois premiers blocs de l'adresse IP soient inversés, puis suivis de l'entité `.in-addr.arpa`. Ceci permet d'associer correctement à cette zone le bloc unique de nombres IP utilisé dans le fichier de zone de résolution de nom inversée.

12.4. Utilisation de `rndc`

BIND contient un utilitaire appelé `rndc` qui permet d'utiliser des lignes de commande pour administrer le démon `named` à partir de l'hôte local ou d'un hôte distant.

Afin d'empêcher l'accès non-autorisé au démon `named`, BIND utilise une méthode de clé secrète partagée pour accorder des privilèges aux hôtes. Dans une telle situation, une clé identique doit être présente aussi bien dans `/etc/named.conf` que dans le fichier de configuration de `rndc`, à savoir `/etc/rndc.conf`

12.4.1. Configuration de `/etc/named.conf`

Pour que `rndc` puisse se connecter à un service `named`, une déclaration `controls` doit être présente dans le fichier `/etc/named.conf` du serveur BIND.

La déclaration `controls` montrée dans l'exemple qui suit, permet à `rndc` de se connecter à partir d'un hôte local.

```
controls {
  inet 127.0.0.1 allow { localhost; } keys { <nom-clé>; };
};
```

Cette déclaration indique à `named` de se mettre à l'écoute du port TCP 953 par défaut de l'adresse inversée et d'autoriser les commandes `rndc` provenant de l'hôte local, si la clé adéquate est présentée. Le *<nom-clé>* fait référence à la déclaration `key`, qui se trouve aussi dans le fichier `/etc/named.conf`. L'exemple suivant illustre une déclaration `key`.

```
key "<nom-clé>" {
  algorithm hmac-md5;
  secret "<valeur-clé>";
};
```

Dans ce cas, la *<valeur-clé>* est une clé HMAC-MD5. Afin de créer des clés HMAC-MD5, utilisez la commande suivante:

```
dnssec-keygen -a hmac-md5 -b <longueur-bits> -n HOST <nom-fichier-clé>
```

Une clé d'au moins 256 bits de long est un bon choix. La bonne clé qui doit être placée dans la zone *<valeur-clé>* se trouve dans *<nom-fichier-clé>*.



Attention

Parce que `/etc/named.conf` ne requiert aucun privilège pour être lu, il est recommandé de placer la déclaration `key` dans un fichier séparé que seul le super-utilisateur (ou root) peut lire et d'utiliser ensuite une déclaration `include` afin de le référencer, comme le montre l'exemple suivant:

```
include "/etc/rndc.key";
```

12.4.2. Configuration de `/etc/rndc.conf`

La déclaration `key` représente la déclaration la plus importante contenue dans `/etc/rndc.conf`.

```
key "<nom-clé>" {
  algorithm hmac-md5;
  secret "<valeur-clé>";
};
```

Les éléments *<nom-clé>* et *<valeur-clé>* doivent être absolument identiques à leurs paramètres contenus dans `/etc/named.conf`.

Pour faire correspondre les clés spécifiés dans le fichier `/etc/named.conf` du serveur cible, ajoutez les lignes suivantes au fichier `/etc/rndc.conf`.

```
options {
  default-server  localhost;
  default-key     "<nom-clé>";
```

```
};
```

Cette commande détermine une clé globale par défaut. Toutefois, la commande `rndc` peut également utiliser différentes clés pour différents serveurs, comme le montre l'exemple suivant:

```
server localhost {
  key  "<nom-clé>";
};
```



Attention

Assurez-vous que seul le super-utilisateur (ou root) ne puisse effectuer des opérations de lecture ou écriture dans le fichier `/etc/rndc.conf`.

12.4.3. Options de ligne de commande

Une commande `rndc` se présente sous le format suivant:

```
rndc <options> <commande> <options-commande>
```

Lors de l'exécution de `rndc` sur un hôte local configuré de façon appropriée, les commandes suivantes sont disponibles:

- `halt` — arrête immédiatement le service `named`.
- `querylog` — Déclenche la journalisation (ou logging) de toutes les requêtes effectuées par des clients vers le présent serveur de noms.
- `refresh` — rafraîchit la base de données du serveur de noms.
- `reload` — recharge les fichiers de zone mais conserve toutes les réponses précédemment placées en cache. Cette command permet également d'opérer des changements sur les fichiers de zone sans perdre toutes les résolutions de nom stockées.

  Si vos changements n'affectent qu'une zone particulière, rechargez seulement une zone en ajoutant le nom de la zone après la commande `reload`.
- `stats` — évacue les statistiques courante de `named` vers le fichier `/var/named/named.stats`.
- `stop` — arrête le serveur de manière nette, en enregistrant préalablement toute mise à jour dynamique et donnée *Incremental Zone Transfers* (*IXFR*).

Dans certaines situations, il sera peut-être nécessaire de passer outre les paramètres par défaut contenue dans le fichier `/etc/rndc.conf`. Les options suivantes sont disponibles:

- `-c <fichier-configuration>` — donne à `rndc` l'instruction d'utiliser un autre fichier de configuration que le fichier par défaut `/etc/rndc.conf`.
- `-p <numéro-port>` — spécifie le numéro de port à utiliser pour la connexion de `rndc`, autre que le port par défaut 953.
- `-s <serveur>` — donne à `rndc` l'instruction d'envoyer la commande vers un autre serveur que celui de l'option `default-server` spécifié dans le fichier de configuration.
- `-y <nom-clé>` — spécifie une clé autre que l'option `default-key` dans le fichier `/etc/rndc.conf`.

Des informations supplémentaires sur ces options sont disponibles dans la page de manuel `rndc`.

12.5. Propriétés avancées de BIND

La plupart des implémentations de BIND utilisent `named` pour fournir un service de résolution de noms ou pour faire autorité pour un domaine ou sous-domaine particuliers. Toutefois, la version 9 de BIND possède aussi un certain nombre de propriétés avancées qui, permettent d'offrir un service DNS plus efficace et plus sécurisé.



Attention

Certaines de ces propriétés avancées, comme DNSSEC, TSIG et IXFR, ne doivent être utilisées que dans les environnements de réseau munis de serveurs de noms qui prennent en charge ces propriétés. Si votre environnement de réseau inclut des serveurs de noms autres que BIND ou des versions de BIND plus anciennes, vérifiez si une propriété avancée est bien prise en charge avant d'essayer de la mettre en oeuvre.

Toutes les propriétés évoquées ici sont décrites en détail dans le *BIND 9 Administrator Reference Manual*. Consultez la Section 12.7.1 pour de plus amples informations.

12.5.1. Améliorations du protocole DNS

BIND supporte les Transferts de zone incrémentaux ('Incremental Zone Transfers' ou IXFR), dans lesquels le serveur de noms esclave ne téléchargera que les portions mises à jour d'une zone modifiée sur un serveur de noms maître. Le processus de transfert standard nécessite que la zone entière soit transférée vers chaque serveur de noms esclave même pour des changements mineurs. Pour des domaines très populaires avec des fichiers de zones très longs et de nombreux serveurs de noms esclaves, IXFR rend la notification et les processus de mise à jour bien moins exigeants en ressources.

Notez que IXFR n'est disponible que si vous utilisez une *mise à jour dynamique*) pour opérer des changements sur les enregistrements de zone maître. to make changes to master zone records. Si vous éditez manuellement des fichiers de zone pour opérer des changements, c'est AXFR qui sera utilisé. Vous trouverez plus d'informations sur les mises à jour dynamiques dans le *BIND 9 Administrator Reference Manual*. Reportez-vous à la Section 12.7.1 pour davantage d'informations.

12.5.2. Vues multiples

En fonction la déclaration `view` dans `named.conf`, BIND peut fournir différentes informations, selon l'identité du demandeur de requête.

Cette option est utilisée essentiellement pour contrôler l'accès à des services DNS ayant des fonctions critiques, en refusant l'accès aux clients externes au réseau local mais en permettant les requêtes des clients internes au réseau local.

La déclaration `view` utilise l'option `match-clients` pour faire correspondre les adresses IP ou des réseaux entiers et leur attribuer des options et des données de zones spéciales.

12.5.3. Sécurité

BIND supporte plusieurs méthodes différentes pour protéger la mise à jour et le transfert de zones, aussi bien sur les serveurs de noms maîtres qu'esclaves:

- *DNSSEC* — abréviation de *DNS SECurity*, cette propriété permet de signer cryptographiquement des zones avec une *clé de zone*.

  De cette façon, on peut vérifier que les informations au sujet d'une zone spécifique proviennent d'un serveur de noms qui les a signées avec une clé privée particulière, du moment que le receveur possède la clé publique de ce serveur de noms.

  La version 9 de BIND prend aussi en charge la méthode de clé publique/privée SIG(0) d'authentification de messages.

- *TSIG* — abréviation de *Transaction SIGnatures*; cette propriété permet d'effectuer un transfert de maître à esclave, mais dont l'autorisation n'est accordée qu'après vérification qu'une clé secrète partagée existe sur le serveur maître et le serveur esclave.

  Cette propriété renforce la méthode d'autorisation de transfert basée sur l'adresse IP standard. Un agresseur devra non seulement accéder à l'adresse IP pour transférer la zone, mais devra aussi connaître la clé secrète.

  La version 9 de BIND prend aussi en charge *TKEY*, qui est une autre méthode de clé secrète partagée pour autoriser les transferts de zone.

12.5.4. IP version 6

La version 9 de BIND peut fournir un service de noms dans des environnements IP version 6 (IPv6) grâce aux enregistrements de zone `A6`.

Si votre environnement de réseau inclut aussi bien des hôtes IPv4 que IPv6, utilisez le démon de résolution très léger `lwresd` sur tous vos clients de réseau. Ce démon est un serveur de noms très efficace, fonctionnant uniquement en cache, qui prend en charge les nouveaux enregistrements `A6` et `DNAME` fonctionnant sous IPv6. Consultez la page de manuel relative à `lwresd` pour plus d'informations.

12.6. Erreurs courantes à éviter

De manière générale, les débutants font fréquemment des erreurs en éditant des fichiers de configuration BIND. Évitez les problèmes suivants:

- *Assurez-vous de bien incrémenter le numéro de série lors de toute modification d'un fichier de zone.*

  Si le numéro de série n'est pas incrémenté, il se peut que votre serveur de noms maître possède les informations nouvelles et correctes, mais les serveurs de noms esclaves ne seront jamais notifiés du changement ou ne tenteront pas de rafraîchir leurs données sur cette zone.

- *Faites attention à bien utiliser ellipses et points-virgules correctement dans le fichier `/etc/named.conf`.*

  L'omission d'un point-virgule ou une ellipse non fermée empêcheront `named` e démarrer.

- *Rappelez-vous de placer des points (`.`) dans les fichiers de zone après tous les FQDN et de les omettre pour les noms d'hôtes.*

  Un point à la fin d'un nom de domaine indique un nom de domaine pleinement qualifié (en d'autres termes, complet). Si le point est omis, `named` attachera le nom de la zone ou la valeur `$ORIGIN` à la suite du nom pour le compléter.

- *Si votre pare-feu cause des problèmes en bloquant les connexions depuis le programme `named` vers d'autres serveurs de noms, vous devrez peut-être éditer son fichier de configuration.*

  La version 9 de BIND utilise par défaut des ports attribués au hasard au-delà de 1024, pour envoyer des requêtes à d'autres serveurs de noms. Toutefois certains pare-feu exigent que tous les serveurs de noms utilisent uniquement le port 53 pour communiquer. Il est possible de forcer `named` à utiliser

le port 53 en ajoutant la ligne suivante à la déclaration cela en ajoutant la ligne suivante options de /etc/named.conf:

```
query-source address * port 53;
```

12.7. Ressources supplémentaires

Les sources d'information suivantes fournissent une documentation supplémentaire sur l'utilisation de BIND.

12.7.1. Documentation installée

- BIND propose une gamme complète de documentation installée couvrant de nombreux sujets, chacun d'eux étant placé dans son propre répertoire thématique:
  - /usr/share/doc/bind-<*numéro-version*>/ — contient un fichier README avec une liste des propriétés les plus récentes.
  - /usr/share/doc/bind-<*numéro-version*>/arm/ — contient les versions HTML et SGML de *BIND 9 Administrator Reference Manual*, qui décrit en détail les ressources nécessaires pour BIND, la façon de configurer différents types de serveurs de noms, d'opérer un équilibrage des charges et d'autres sujets avancés. Pour la plupart des nouveaux utilisateurs de BIND, ces ressources constituent le meilleur point de départ.
  - /usr/share/doc/bind-<*numéro-version*>/draft/ — contient des documents techniques assortis qui traite des problèmes en relation avec service DNS et propose quelques solutions pour les résoudre.
  - /usr/share/doc/bind-<*numéro-version*>/misc/ — contient des documents préparés pour aborder des problèmes spécifiques avancés. Les utilisateurs de la version 8 de BIND devraient consulter le document migration pour s'informer des changements importants à faire pour passer à la version 9 de BIND. Le fichier options énumère toutes les options implémentées dans BIND 9, qui sont utilisées dans /etc/named.conf.
  - /usr/share/doc/bind-<*numéro-version*>/rfc/ — tous les documents RFC concernant BIND sont placés dans ce répertoire.

- man named — examine les arguments assortis qui peuvent être utilisés pour contrôler le démon du serveur de noms BIND.
- man named.conf — une liste exhaustive des options disponibles au sein du fichier de configuration named.
- man rndc — explique les différentes options disponibles lors de l'utilisation de la commande rndc pour contrôler un serveur de noms BIND.
- man rndc.conf — une liste exhaustive des options disponibles au sein du fichier de configuration rndc.

12.7.2. Sites Web utiles

- http://www.isc.org/products/BIND — La page d'accueil du projet BIND, où vous pourrez trouver des informations sur les versions actuelles ainsi qu'une version PDF de *BIND 9 Administrator Reference Manual*.
- http://www.redhat.com/mirrors/LDP/HOWTO/DNS-HOWTO.html — Couvre l'utilisation de BIND en tant que serveur de noms de résolution en cache, ou bien la configuration de divers

fichiers de zone nécessaires pour qu'il soit utilisé comme serveur de noms primaire pour un domaine.

12.7.3. Livres sur le sujet

- *DNS and BIND* de Paul Albitz et Cricket Liu; publié par O'Reilly & Associates — Un livre de référence populaire qui explique les options de configuration de BIND des plus simples aux plus ésotériques, et fournit aussi des stratégies pour sécuriser votre serveur DNS.
- *The Concise Guide to DNS and BIND* de Nicolai Langfeldt; publié par Que — Examine la connexion entre les services de réseaux multiples et BIND, en mettant l'accent sur les sujets techniques et orientés vers des applications pratiques.



Chapitre 13. Protocole LDAP (Lightweight Directory Access Protocol)

Lightweight Directory Access Protocol (*LDAP*) est un ensemble de protocoles ouverts utilisés pour accéder à des informations stockées localement sur un réseau. Il est basé sur le standard *X.500* pour le partage de répertoires, mais est moins complexe et exigeant en matière de ressource. C'est pour cette raison que LDAP est quelques fois appelé "*X.500 Lite*."

Comme X.500, LDAP organise des informations d'une manière hiérarchique en utilisant des répertoires. Ces répertoires peuvent stocker diverses informations et peuvent même être utilisés d'une manière semblable au Service d'informations réseau(NIS, Network Information Service), permettant à tout un chacun d'accéder à son compte depuis toute machine dans le réseau sous LDAP.

Dans la plupart des cas, cependant, LDAP sert seulement d'annuaire téléphonique virtuel, permettant aux utilisateurs d'accéder facilement aux informations de contact d'autres utilisateurs. Mais le protocole LDAP est beaucoup plus utile qu'un annuaire papier. En effet, de par sa conception, il est destiné à prendre en charge la propagation vers des serveurs LDAP sur tout l'Internet, fournissant ainsi un accès mondial aux informations. Actuellement, le protocole LDAP est plus généralement utilisé au sein de grandes organisations comme des universités, des départements gouvernementaux et entreprises du secteur privé.

Le protocole LDAP est un système client/serveur. Le serveur peut utiliser diverses bases de données pour stocker un répertoire, chacune d'elles étant optimisée de façon à permettre des opérations de consultation rapides et nombreuses. Lorsqu'un client LDAP se connecte à un serveur LDAP il peut soit consulter un répertoire, soit y apporter des modifications. Dans le cas d'une requête, le serveur y répond, ou, s'il ne peut pas le faire localement, la renvoie à un serveur LDAP de niveau supérieur qui aura lui la réponse. Si l'application cliente tente de changer des informations dans un répertoire LDAP, le serveur vérifie d'abord que l'utilisateur est bien autorisé à effectuer des changement et ensuite ajoute ou met à jour les informations.

Ce chapitre décrit la configuration et l'utilisation de OpenLDAP 2.0, une implémentation Open Source des protocoles LDAPv2 et LDAPv3.

13.1. Pourquoi utiliser LDAP?

Le principal avantage du protocole LDAP réside dans la possibilité de réunir les informations d'une organisation entière dans un lieu central. Par exemple, toutes les listes d'utilisateurs au sein de l'organisation peuvent être fusionnées dans un répertoire LDAP. Ce répertoire peut être interrogé par toute application compatible avec LDAP ayant besoin de ces informations. De plus, puisque LDAP supporte les fonctions Secure Sockets Layer (SSL) et Transport Layer Security (TLS), des données sensibles peuvent être protégées des intrusions.

LDAP supporte aussi diverses bases de données parallèles pour y enregistrer des répertoires. Cela donne aux administrateurs la flexibilité nécessaire pour déployer la base de données la plus adaptée au type d'informations que le serveur doit disséminer. De plus, comme LDAP comporte une API (de l'anglais 'Application Programming Interface', soit interface de programmation d'application) bien définie, le nombre d'applications compatibles avec LDAP est vaste et grandissant aussi bien en quantité qu'en qualité.

Du côté négatif, LDAP peut être difficile à configurer.

13.1.1. Améliorations des caractéristiques d'OpenLDAP 2.0

OpenLDAP 2.0 comprend plusieurs caractéristiques importantes.

- *Support LDAPv3* — OpenLDAP 2.0 supporte SASL ('Simple Authentication and Security Layer'), TLS ('Transport Layer Security'), et SSL ('Secure Sockets Layer'), entre autres améliorations. De nombreux changements apportés au protocole depuis LDAPv2 visent à augmenter la sécurité de LDAP.
- *Support IPv6* — OpenLDAP supporte le protocole Internet de la prochaine génération, version 6.
- *LDAP sur IPC* — OpenLDAP peut communiquer au sein d'un système en utilisant IPC ('interprocess communication'). Il en résulte une sécurité améliorée car il n'est plus nécessaire de communiquer à travers un réseau.
- *Mise à jour de C API* — Elle améliore la manière dont les programmeurs se connectent aux serveurs de répertoires LDAP et les utilisent.
- *Support LDIFv1* — Grâce à ce support, OpenLDAP 2.0 est pleinement compatible avec la version 1 du format d'échange de données LDAP (LDIF).
- *Amélioration du serveur autonome LDAP* — OpenLDAP inclue à présent un système de contrôle d'accès mis à jour, un pool de conversation, de meilleurs outils et bien plus encore.

13.2. Terminologie de LDAP

Toute discussion de LDAP nécessite une compréhension de base d'un certain nombre de termes spécifiques à LDAP:

- *entrée* — correspond à une seule unité dans un répertoire LDAP. Chaque entrée est identifiée ou référencée par son *Nom distinctif ou DN* (de l'anglais 'Distinguished Name') unique.
- *attributs* — Des attributs sont des éléments d'information directement associés à l'entrée. Par exemple, une organisation pourrait être représentée par une entrée LDAP. Parmi les attributs associés à l'organisation on pourrait avoir son numéro de fax, son adresse, etc. Des personnes pourraient également constituer des entrées dans le répertoire LDAP. Parmi les attributs courants utilisés pour les personnes figurent les numéros de téléphone et adresses électroniques.

  Certains attributs sont obligatoires, tandis que d'autres sont facultatifs. Une *classe d'objets* définit les attributs obligatoires et les attributs facultatifs. Vous trouverez des définitions de classes d'objets dans différents fichiers schéma placés dans le répertoire `/etc/openldap/schema/`. Pour de plus amples informations sur le schéma LDAP, consultez la Section 13.5.
- *LDIF* — Le *LDAP Data Interchange Format* (LDIF: format d'échange de données LDAP) est un format de texte ASCII pour les entrées LDAP. Les fichiers qui échangent des données avec des serveurs LDAP doivent être de format LDIF. Une entrée LDIF ressemble à l'extrait ci-dessous:

```
[<id>]
dn: <distinguished name>
<attrtype>: <attrvalue>
<attrtype>: <attrvalue>
<attrtype>: <attrvalue>
```

  Toute entrée peut contenir autant de paires `<attrtype>: <attrvalue>` que nécessaire. Une ligne vierge indique que l'entrée est terminée.


Attention

Toutes les paires <*attrtype*> et <*attrvalue*> *doivent* être définies par un fichier de schéma avant de pouvoir utiliser ces informations.

Toute élément contenu entre "<" et ">" est une variable que vous pouvez paramétrer lorsque vous ajoutez une entrée LDAP. Toutefois, ce n'est pas le cas de <*id*>. Cet élément <*id*> est un nombre paramétré par l'application utilisée lors de l'ajout d'une entrée.



Remarque

Il est fortement conseillé de ne jamais éditer manuellement une entrée LDIF. Utilisez plutôt les applications clientes LDAP, comme celles énumérées dans la Section 13.3.

13.3. Démons et utilitaires OpenLDAP

Cette suite de bibliothèques et d'outils OpenLDAP et répartie dans les paquetages suivants:

- `openldap` — Contient les bibliothèques nécessaires pour faire fonctionner le serveur OpenLDAP et les applications clientes.
- `openldap-clients` — Contient les outils de ligne de commande pour visualiser et modifier les répertoires d'un serveur LDAP.
- `openldap-servers` — Contient les serveurs et autres utilitaires nécessaires pour configurer et faire fonctionner un serveur LDAP.

Deux serveurs sont contenus dans le paquetage `openldap-servers`: le *démon autonome LDAP* (`/usr/sbin/slapd`) et le *démon autonome LDAP de réplication de mise à jour* (`/usr/sbin/slurpd`).

Le démon `slapd` est un serveur LDAP autonome, tandis que le démon `slurpd` sert à synchroniser les changements d'un serveur LDAP vers les autres serveurs LDAP du réseau. Le démon `slurpd` n'est nécessaire que pour un serveur LDAP multiple.

Pour effectuer des tâches administratives, le paquetage `openldap-servers` installe les utilitaires suivants dans le répertoire `/usr/sbin/`:

- `slapadd` — Ajoute des entrées d'un fichier LDIF vers un répertoire LDAP. Par exemple, la commande `/usr/sbin/slapadd -l` *`ldif-input`* lira le fichier LDIF *`ldif-input`* contenant les nouvelles entrées.
- `slapcat` — Extrait des données d'un répertoire LDAP dans le format par défaut — Berkeley DB — et les enregistre dans un fichier LDIF. Par exemple, la commande `/usr/sbin/slapcat -l` *`ldif-output`* produira un fichier LDIF nommé *`ldif-output`* qui contient les entrées du répertoire LDAP.
- `slapindex` — Indexe à nouveau le répertoire `slapd` à partir du contenu actuel.
- `slappasswd` — Crée une valeur pour le mot de passe utilisateur à utiliser avec `ldapmodify` ou la valeur `rootpw` dans le fichier de configuration `slapd`, `/etc/openldap/slapd.conf`. Exécutez la commande `/usr/sbin/slappasswd` pour créer le mot de passe.



Avertissement

Assurez-vous d'avoir arrêté `slapd` par la commande `/usr/sbin/service slapd stop` avant d'utiliser `slapadd`, `slapcat` ou `slapindex`. Sinon vous risquez d'endommager votre répertoire LDAP.

Pour plus d'informations sur l'utilisation de ces outils, consultez les pages de manuel qui y sont consacrées.

Le paquetage `openldap-clients` installe dans `/usr/bin/` des outils permettant d'ajouter, modifier et supprimer des entrées dans un répertoire LDAP. Parmi ces outils se trouvent:

- `ldapmodify` — Modifie les entrées dans un répertoire LDAP, acceptant leur apport par un fichier ou par inscription standard.
- `ldapadd` — Ajoute des entrées dans votre répertoire, acceptant leur apport par un fichier ou par inscription standard; `ldapadd` est en fait un lien dur vers la commande `ldapmodify -a`.
- `ldapsearch` — Recherche des entrées dans un répertoire LDAP par une invite du shell.
- `ldapdelete` — Supprime des entrées dans un répertoire LDAP en acceptant l'action dont l'origine est un fichier ou l'utilisateur au moyen du terminal.

À l'exception de la commande `ldapsearch`, chacun de ces utilitaires a une utilisation plus facile en fonctionnant par référence à un fichier contenant les changements à effectuer plustôt que par l'utilisation d'une commande pour chaque entrée que vous désirer changer dans le répertoire LDAP. Le format d'un tel fichier est expliqué dans les pages de manuel relative à chaque application.

13.3.1. NSS, PAM et LDAP

Outre les paquetages OpenLDAP, Red Hat Linux comprend un paquetage nommé `nss_ldap` qui améliore la capacité de LDAP à s'intégrer aussi bien dans un environnement Linux que tout autre environnement UNIX.

Le paquetage `nss_ldap` fournit les modules suivants:

- `/lib/libnss_ldap-<glibc-version>.so`
- `/lib/security/pam_ldap.so`

Le module `libnss_ldap-<glibc-version>.so` permet aux applications de rechercher les utilisateurs, les groupes, les hôtes et d'autres informations en utilisant un répertoire LDAP via l'interface glibc *Nameservice Switch* (NSS). NSS permet l'authentification d'applications en utilisant LDAP avec le service de noms *Network Information Service* (NIS) et les fichiers simples pour l'authentification.

Le module `pam_ldap` permet aux applications fonctionnant avec PAM d'authentifier les utilisateurs en utilisant les informations stockées dans un répertoire LDAP. Les applications fonctionnant avec PAM comprennent le login de console, les serveurs de mail POP et IMAP, et Samba. En déployant un serveur LDAP sur votre réseau, toutes ces applications peuvent, pour leur authentification, utiliser la même combinaisons nom d'utilisateur/mot de passe, ce qui simplifie grandement l'administration.

13.3.2. PHP4, Serveur HTTP Apache, et LDAP

Red Hat Linux comprend aussi des paquetages avec des modules LDAP pour le Serveur HTTP Apache et le langage de scripte PHP côté serveur.

Le paquetage `php-ldap` ajoute le support LDAP au langage de script PHP4 à HTML intégré grâce au module `/usr/lib/php4/ldap.so`. Ce module permet aux scripts PHP4 d'accéder aux informations stockées dans une répertoire LDAP.



Important

Red Hat Linux n'inclut plus le paquetage `auth_ldap`. Ce dernier fournissait le support du Serveur HTTP Apache version 1.3 et versions précédentes. Pour toute informations concernant le statut de ce module, rendez-vous sur le site Web de Apache Software Foundation à l'adresse suivante: http://www.apache.org/.

13.3.3. Applications clientes LDAP

Il existe des clients LDAP graphiques qui supportent la création et la modification de répertoires, mais ces applications ne sont pas incluses dans Red Hat Linux. Un exemple est le navigateur/éditeur **LDAP Browser/Editor** — Cet outil basé sur Java est disponible en ligne à l'adresse suivante: http://www.iit.edu/~gawojar/ldap.

La plupart des autres clients LDAP accèdent aux répertoires en lecture seulement et les utilisent pour référencer, et non pas modifier, les informations de l'entreprise en général. Parmi ces applications, on compte les navigateurs Web basés sur Mozilla, Sendmail, **Balsa**, **Pine**, **Evolution** et **Gnome Meeting**.

13.4. Fichiers de configuration OpenLDAP

Les fichiers de configuration OpenLDAP sont installés dans le répertoire `/etc/openldap/`. Ci-dessous figure une brève liste des répertoires et fichiers les plus importants:

- `/etc/openldap/ldap.conf` — Ce fichier est le fichier de configuration pour toutes les applications *clientes* qui utilisent les bibliothèques comme `ldapsearch`, `ldapadd`, Sendmail, **Pine**, **Balsa**, **Evolution**, et **Gnome Meeting**.
- Le répertoire `/etc/openldap/slapd.conf` — Ce fichier de configuration est celui du démon `slapd`. Pour plus d'informations sur ce fichier, reportez-vous à la Section 13.6.1.
- `/etc/openldap/schema/` — Ce sous-répertoire contient le schéma utilisé par le démon `slapd`. Pour plus d'informations sur ce répertoire, reportez-vous à la Section 13.5.



Remarque

Si le paquetage `nss_ldap` est installé, il créera un fichier nommé `/etc/ldap.conf`. Ce fichier est utilisé par les modules PAM et NSS fournis par le paquetage `nss_ldap`. Pour de plus amples informations sur ce fichier de configuration, consultez la Section 13.7.

13.5. Le répertoire `/etc/openldap/schema/`

Le répertoire `/etc/openldap/schema/` contient les définitions de LDAP précédemment placées dans les fichiers `slapd.at.conf` et `slapd.oc.conf`. Toutes les *définitions de syntaxe d'attribut* et *définitions de la classe d'objet* sont maintenant placées dans des fichiers schéma différents. Ces derniers sont référencés dans `/etc/openldap/slapd.conf` en utilisant les lignes `include`, comme dans l'exemple ci-dessous:

```
include  /etc/openldap/schema/core.schema
include  /etc/openldap/schema/cosine.schema
include  /etc/openldap/schema/inetorgperson.schema
include  /etc/openldap/schema/nis.schema
include  /etc/openldap/schema/rfc822-MailMember.schema
include  /etc/openldap/schema/autofs.schema
include  /etc/openldap/schema/kerberosobject.schema
```



Attention

Vous ne devriez modifier aucun élément du schéma défini dans les fichiers schéma installés par OpenLDAP.

Ceci étant, vous pouvez étendre le schéma utilisé par OpenLDAP afin de supporter d'autres types d'attributs et classes d'objets en utilisant comme guide, les fichiers schéma par défaut. Pour ce faire, créez un fichier `local.schema` dans le répertoire `/etc/openldap/schema`. Référencez ce nouveau schéma dans `slapd.conf` en ajoutant les lignes suivantes en dessous de vos lignes de schéma `include` par défaut:

```
include          /etc/openldap/schema/local.schema
```

Ensuite, définissez Next, vos nouveaux types d'attributs et classes d'objets dans le fichier `local.schema`. Beaucoup d'organisations utilisent les types d'attributs et classes d'objet existants dans les fichiers de schéma installés par défaut et les modifient pour usage dans le fichier `local.schema` file.

Étendre un schéma en fonction de besoins spécialisés est une tâche complexe qui dépasse le cadre du présent chapitre. Consultez http://www.openldap.org/doc/admin/schema.html pour plus d'informations sur l'écritures de nouveaux fichiers de schéma.

13.6. Aperçu de la configuration de OpenLDAP

Cette section fournit une présentation rapide des opérations à accomplir pour installer et configurer un annuaire OpenLDAP. Pour plus d'informations, reportez-vous aux URL suivantes:

- http://www.openldap.org/doc/admin/quickstart.html — Le *Quick-Start Guide* sur le site Web d'OpenLDAP.
- http://www.redhat.com/mirrors/LDP/HOWTO/LDAP-HOWTO.html — Le *LDAP Linux HOWTO* du Projet de documentation Linux, en miroir sur le site Red Hat.

Ci-dessous figurent les étapes de base pour créer un serveur LDAP:

1. Installez les RPM de `openldap`, `openldap-servers`, et `openldap-clients`.
2. Éditez le fichier `/etc/openldap/slapd.conf` afin de référencer votre domaine ainsi que votre serveur LDAP. Reportez-vous à la Section 13.6.1 afin d'obtenir davantage d'informations sur la manière d'éditer ce fichier.
3. Lancez `slapd` à l'aide de la commande:
   ```
   /sbin/service/ldap start
   ```
   Après avoir correctement configuré LDAP, vous pouvez utiliser `chkconfig`, `ntsysv`, ou l'**Outil de configuration des services** pour configurer LDAP de façon à le lancer avec le système. Pour de plus amples informations sur la configuration des services, consultez le chapitre intitulé *Contrôle de l'accès aux services* du *Guide de personnalisation de Red Hat Linux*.
4. Ajoutez des entrées à votre répertoire LDAP à l'aide de `ldapadd`.
5. Utilisez `ldapsearch` afin de vérifier si `slapd` accède correctement aux informations.
6. À ce stade, votre répertoire LDAP devrait exister. L'étape suivante consiste à configurer vos applications compatibles avec LDAP de manière à ce qu'elles puissent utiliser le répertoire LDAP.

13.6.1. Édition de /etc/openldap/slapd.conf

Afin d'utiliser le serveur LDAP slapd, vous devrez modifier son fichier de configuration, /etc/openldap/slapd.conf. Vous devez éditer le fichier de façon à spécifier le domaine et le serveur corrects.

La ligne de suffix nomme le domaine pour lequel le serveur LDAP fournira les informations et devrait être changée ainsi:

```
suffix          "dc=your-domain,dc=com"
```

de façon à refléter votre nom de domaine. Par exemple:

```
suffix          "dc=example,dc=com"
```

L'entrée rootdn est le *Nom distinctif* (*DN*) pour un utilisateur non restreint par les paramètres de contrôle d'accès ou de limite administrative définis pour des opérations sur le répertoire LDAP. L'utilisateur rootdn peut être considéré comme le super-utilisateur pour le répertoire LDAP. Dans le fichier de configuration, changez la ligne rootdn de sa valeur par défaut à quelquechose semblable à la ligne ci-dessous:

```
rootdn          "cn=root,dc=example,dc=com"
```

Si vous avez l'intention de remplir le répertoire LDAP sur le réseau, modifiez la ligne rootpw — en remplaçant la valeur par défaut par une chaîne de mot de passe cryptée. Afin de créer une chaîne de mots de passe cryptée, tapez la commande suivante:

```
slappasswd
```

Il vous sera demandé d'inscrire et de réinscrire un mot de passe. Le programme imprime ensuite le mot de passe crypté vers le terminal.

Ensuite, copiez le mot de passe crypté que vous venez de créer dans >/etc/openldap/slapd.conf sur une des lignes rootpw et supprimez le signe dièse (#).

Une fois cette modification apportée, la ligne devrait ressembler à l'exemple ci-dessous:

```
rootpw {SSHA}vv2y+i6V6esazrIv70xSSnNAJE18bb2u
```



Avertissement

Les mots de passe LDAP, y compris la directive rootpw spécifiée dans /etc/openldap/slapd.conf, sont envoyés sur le réseau en *texte simple*, à moins que vous ne permettiez le cryptage TLS.

Pour permettre le cryptage TLS, passez en revue les commentaires figurant dans /etc/openldap/slapd.conf et consultez la page de manuel relative à slapd.conf.

Pour une meilleure sécurité, la directive rootpw devrait être commentée après avoir peuplé le répertoire LDAP. Pour ce faire, ajoutez un signe dièse avant cette directive (#).

Si vous utilisez l'outil de ligne de commande /usr/sbin/slapadd localement pour peupler le répertoire, il n'est pas nécessaire d'utiliser la directive rootpw.



Important

Vous devez être connecté en tant que super-utilisateur pour pouvoir utiliser `/usr/sbin/slapadd`. Toutefois, le serveur de répertoires tourne en tant que l'utilisateur `ldap`. Par conséquent, le serveur de répertoires ne sera pas en mesure de modifier tout fichier créé par `slapadd`. Pour résoudre ce problème, tapez la commande ci-dessous lorsque vous avez fini d'utiliser `slapadd`:

```
chown -R ldap /var/lib/ldap
```

13.7. Configuration de votre système pour l'authentification à l'aide de OpenLDAP

Cette section donne un bref aperçu de la manière de configurer votre système Red Hat Linux pour permettre l'authentification à l'aide de OpenLDAP. À moins que vous ne soyez un expert de OpenLDAP, vous aurez probablement besoin de plus de documentation que vous n'en trouverez ici. Reportez-vous aux références de la Section 13.9 pour de plus amples informations.

Installez les paquetages LDAP nécessaires

Commencez par vérifier que les paquetages appropriés sont présents à la fois sur le serveur LDAP et sur les machines LDAP clientes. Le serveur LDAP nécessite le paquetage `openldap-servers`.

Les paquetages `openldap`, `openldap-clients` et `nss_ldap` doivent être installés sur tous les ordinateurs clients LDAP.

Éditez les fichiers de configuration

- Sur le serveur LDAP, éditez le fichier `/etc/openldap/slapd.conf` pour vous assurer qu'il correspond bien aux élément spécifiques de votre organisation. Pour obtenir des instructions sur la manière d'éditer `slapd.conf` reportez-vous à la Section 13.6.1.
- Sur les ordinateurs clients, `/etc/ldap.conf` et `/etc/openldap/ldap.conf` doivent contenir les informations correctes sur le serveur et la base de recherche de votre organisation.

  La façon la plus simple de procéder consiste à lancer l'**Outil de configuration d'authentification** (`authconfig-gtk`) et à sélectionner **Activer le support LDAP** sous l'onglet **Informations utilisateur**.

  Vous pouvez aussi éditer ces fichiers manuellement.
- Sur les ordinateurs clients, les fichier `/etc/nsswitch.conf` doit être édité afin de pouvoir utiliser LDAP.

  Pour ce faire, la façon la plus simple consiste à lancer l'**Outil de configuration d'authentification** (`authconfig-gtk`) et à sélectionner **Activer le support LDAP** sous l'onglet **Informations utilisateur**.

  Si vous éditez `/etc/nsswitch.conf` manuellement, ajoutez `ldap` aux lignes appropriées.

  Comme par exemple:
  ```
  passwd: files ldap
  shadow: files ldap
  group: files ldap
  ```

13.7.1. PAM et LDAP

Pour faire en sorte que des applications compatibles avec PAM standard utilisent LDAP pour l'authentification, exécutez **Outil de configuration d'authentification** et sélectionnez **Activer le support LDAP** sous l'onglet **Authentification**. Pour de plus amples informations sur la configuration de PAM, consultez le Chapitre 14 et les pages de manuel relatives à PAM.

13.7.2. Migration de vos anciennes informations d'authentification vers le format LDAP

Le répertoire `/usr/share/openldap/migration/` contient un ensemble de scripts shell et Perl pour la migration de vos anciennes informations d'authentification vers le format LDAP.

Tout d'abord, modifiez le fichier `migrate_common.ph` de manière à ce qu'il reflète votre domaine. Le domaine DNS par défaut devrait être changé pour ressembler à ceci:

```
$DEFAULT_MAIL_DOMAIN = "votre_société";
```

La base par défaut devrait également être changée, pour ressembler à ceci:

```
$DEFAULT_BASE = "dc=votre_organisation,dc=com";
```

Le travail de migration d'une base de données d'utilisateur vers un format lisible par LDAP incombe à un groupe de scripts de migration installés dans le même répertoire. À l'aide du Tableau 13-1, déterminez le script à utiliser pour la migration de votre base de données d'utilisateur.

Service de noms existant	LDAP fonctionne-t-il?	Script à utiliser
`/etc` flat files	oui	`migrate_all_online.sh`
`/etc` flat files	non	`migrate_all_offline.sh`
NetInfo	oui	`migrate_all_netinfo_online.sh`
NetInfo	non	`migrate_all_netinfo_offline.sh`
NIS (YP)	oui	`migrate_all_nis_online.sh`
NIS (YP)	non	`migrate_all_nis_offline.sh`

Tableau 13-1. Scripts de migration LDAP

Exécutez le script approprié en fonction de votre service de noms existant.



Remarque

Perl doit être installé sur votre système pour que vous puissiez utiliser ces scripts.

Les fichiers `README` et `migration-tools.txt` du répertoire `/usr/share/openldap/migration/` fournissent plus de détails sur la migration d'informations.

13.8. Mise à niveau pour une Version 2.0 de OpenLDAP

Dans la Version 2.0 de OpenLDAP, le format de stockage sur disque utilisé par le serveur `slapd` LDAP est différent. Si vous faites une mise à niveau de LDAP à partir de Red Hat Linux 7.0 ou une version antérieure, vous devrez extraire les répertoires LDAP existants pour les placer dans un fichier LDIF à l'aide de la commande suivante:

```
ldbmcat -n > <ldif_file>
```

Dans la commande ci-dessus, remplacez `<ldif_file>` par le nom du fichier de sortie. Tapez ensuite la commande suivante pour importer ce fichier dans OpenLDAP 2.0:

```
slapadd -l <ldif_file>
```



Important

Vous devez être connecté en tant que super-utilisateur pour pouvoir utiliser `/usr/sbin/slapadd`. Toutefois, le serveur de répertoires tourne en tant que l'utilisateur `ldap`. Par conséquent, le serveur de répertoires ne sera pas en mesure de modifier tout fichier créé par `slapadd`. Pour résoudre ce problème, tapez la commande ci-dessous lorsque vous avez fini d'utiliser `slapadd`:

```
chown -R ldap /var/lib/ldap
```

13.9. Ressources supplémentaires

Il existe d'autres informations concernant LDAP. Consultez ces sources, en particulier le site Web OpenLDAP et le HOWTO LDAP, avant de commencer à configurer LDAP sur votre système.

13.9.1. Documentation installée

- La page du manuel relative à LDAP — La page de manuel de `ldap` constitue un bon point de départ pour une introduction à LDAP. Vous trouverez aussi des pages de manuel consacrées aux démons et utilitaires de LDAP.
- `/usr/share/docs/openldap-<version-number>` — Contient un document `README` général ainsi que des informations diverses.

13.9.2. Sites Web utiles

- http://www.openldap.org/ — Site du projet OpenLDAP. Ce site Web contient de nombreuses informations sur la configuration de OpenLDAP.
- http://www.redhat.com/mirrors/LDP/HOWTO/LDAP-HOWTO.html — Un document LDAP HOWTO plus ancien mais toujours pertinent.
- http://www.padl.com/ — Les développeurs de `nss_ldap` et `pam_ldap` entre autres outils LDAP utiles.
- http://www.kingsmountain.com/ldapRoadmap.shtml — La Road Map LDAP de Jeff Hodges contient des liens vers différents Forums aux questions et des nouvelles importantes concernant le protocole LDAP.

- http://www.webtechniques.com/archives/2000/05/wilcox — Un regard utile sur la gestion des groupes dans LDAP.
- http://www.ldapman.org/articles — Articles offrant une bonne introduction à LDAP, ainsi que des méthodes de création d'arborescence de répertoires et des structures de répertoire de personnalisation.

13.9.3. Livres sur le sujet

- *Implementing LDAP* de Mark Wilcox; Wrox Press, Inc.
- *Understanding and Deploying LDAP Directory Services* de Tim Howes et al.; Macmillan Technical Publishing

III. Références à la sécurité

L'utilisation de protocoles sécurisés représente un élément vital dans le maintien de l'intégrité d'un système. Cette partie se concentre sur certains outils critiques utilisés pour l'identification des utilisateurs, le contrôle de l'accès au réseau, l'établissement de communications réseau sécurisées et la détection des intrusions. Pour obtenir de plus amples informations sur la sécurisation d'un système Red Hat Linux, reportez-vous au *Guide de sécurité de Red Hat Linux*.

Table des matières



Chapitre 14. Modules d'authentification enfichables (PAM)

Des programmes qui permettent à des utilisateurs d'accéder à un système vérifient préalablement l'identité de l'utilisateur au moyen d'un processus d'*authentification*. Dans le passé, chaque programme de ce genre effectuait les opérations d'authentification d'une manière qui lui était propre. Sous Red Hat Linux, un grand nombre de ces programmes sont configurés de telle sorte qu'ils utilisent un processus d'authentification centralisé appelé *modules d'authentification enfichables* (ou *PAM* de l'anglais 'Pluggable Authentication Modules').

PAM utilise une architecture modulaire enfichable, offrant à l'administrateur système une grande flexibilité quant à l'établissement d'une politique d'authentification pour le système.

Dans la plupart des cas, vous n'aurez pas à modifier les fichiers de configuration PAM par défaut pour les applications qui prennent en charge les PAM. Toutefois, il sera parfois nécessaire dans certains cas de modifier le fichier un configuration PAM. Étant donné qu'une mauvaise configuration de PAM peut compromettre la sécurité de votre système, il est important de comprendre la structure de ces fichiers avant de leur apporter toute modification (reportez-vous à la Section 14.3 pour de plus amples informations).

14.1. Avantages des PAM

PAM offre entre autres les avantages suivants:

- il fournit un système d'authentification commun qui pouvant être utilisé avec un vaste éventail d'applications;
- il offre un haut degrés de flexibilité et de contrôle en ce qui concerne l'authentification aussi bien au niveau de l'administrateur système qu'au niveau du développeur d'applications;
- il permet aux développeurs d'applications de concevoir des programmes sans avoir à créer leur propre système d'authentification.

14.2. Fichiers de configuration PAM

Le répertoire `/etc/pam.d/` contient les fichiers de configuration PAM pour les applications prenant en charge les PAM. Les versions précédentes de PAM utilisaient le fichier `/etc/pam.conf`, mais ce dernier a été abandonné et `pam.conf` est lu seulement si le répertoire `/etc/pam.d/` n'existe pas.

14.2.1. Fichiers de services PAM

Chaque application ou *service* prenant en charge les PAM correspond à un fichier dans le répertoire `/etc/pam.d/`. Chacun de ces fichiers est nommé en fonction du service dont il contrôle l'accès.

Il appartient au programme prenant en charge les PAM de définir le nom de ses services et d'installer son fichier de configuration PAM dans le répertoire `/etc/pam.d/`. Par exemple, le programme `login` attribue le nom `/etc/pam.d/login` à son service .

14.3. Format des fichiers de configuration PAM

Chaque fichier de configuration PAM comprend un ensemble de directives établies selon format suivant:

<*interface-module*> <*indicateur-contrôle*> <*chemin-module*> <*arguments-module*>

Les sections suivantes décrivent ces éléments un par un.

14.3.1. Interface du module

Il existe quatre types d’interface pour les modules PAM, chacune correspondant à un aspect différent du processus d’autorisation:

- `auth` — Ces modules sont utilisés pour authentifier l’utilisateur, par exemple en lui demandant son mot de passe et en le vérifiant. Les modules avec cette peuvent également établir des certificats d’identité, tels qu’une inscription à un groupe ou des tickets Kerberos.
- `account` — Ces modules sont utilisés pour vérifier que l’accès est bien autorisé. Par exemple, ils peuvent vérifier si le compte a expiré ou non, ou bien si l’utilisateur est autorisé à se connecter à un moment donné de la journée.
- `password` — Ces modules sont utilisés pour définir les mots de passe.
- `session` — Ces modules sont utilisés pour configurer et gérer des sessions d’utilisateurs. These modules configure and manage user sessions. Les modules ayant cette interface peuvent également effectuer des tâches supplémentaires requises pour autoriser l’accès, comme par exemple pour monter le répertoire personnel d’un utilisateur ou activer sa boîte aux lettres.



Remarque

Un module individuel peut fournir une interface de module particulière ou toutes les interfaces de modules. Par exemple, `pam_unix.so` fournit les quatre interfaces.

Dans un fichier de configuration PAM, l’interface de module est le premier aspect a être défini. Par exemple, une ligne typique d’une configuration pourrait ressembler à l’extrait suivant:

```
auth      required  /lib/security/pam_unix.so
```

Cette ligne donne l’instruction aux PAM d’utiliser l’interface `auth` du module `pam_unix.so`.

14.3.1.1. Modules d’empilage

Les directives des interfaces de modules peuvent être *empilées* ou placées les une sur les autres, afin que de multiples modules puissent être utilisés ensemble dans un but particulier. Dans de telles circonstances, l’ordre dans lequel les modules sont répertoriés est très important au niveau du processus d’authentification.

Grâce à l’empilage, un administrateur peut facilement exiger la présence de différentes conditions avant d’autoriser un utilisateur à s’authentifier. Par exemple, `rlogin` utilise normalement cinq modules `auth` empilés, comme le montre son fichier de configuration PAM:

```
auth       required     /lib/security/pam_nologin.so
auth       required     /lib/security/pam_securetty.so
auth       required     /lib/security/pam_env.so
auth       sufficient   /lib/security/pam_rhosts_auth.so
auth       required     /lib/security/pam_stack.so service=system-auth
```

Avant d’accorder l’autorisation d’utilisation de `rlogin`, PAM s’assure que le fichier `/etc/nologin` n’existe pas, que l’utilisateur n’essaie pas de se connecter à distance en tant que super-utilisateur (ou root) au moyen d’une connexion réseau non-cryptée et que toutes les variables d’environnement peuvent être chargées. Si une authentification `rhosts` peut être établie avec succès, la connexion est alors autorisée. En revanche, si l’authentification `rhosts` échoue, une authentification standard de mot de passe est exécutée.

14.3.2. Indicateurs de contrôle

Lorsqu’ils sont appelés, tous les modules PAM donnent un résultat indiquant soit la réussite, soit l’échec. Les indicateurs de contrôle indiquent aux PAM comment traiter ce résultat. Étant donné que les modules peuvent être empilés dans un ordre bien précis, les indicateurs de contrôle décident de l’importance de la réussite ou de l’échec d’un module spécifique par rapport au but général d’authentification d’un utilisateur pour un service donné.

Il existe quatre types d’indicateurs de contrôle prédéfinis, à savoir:

- `required` — le module doit être vérifié avec succès pour que l’authentification puise se poursuivre. Si la vérification d’un module portant l’indication `required` échoue, l’utilisateur n’en est pas averti tant que tous les modules associés à cette interface n’ont pas été vérifiés.
- `requisite` — le module doit être vérifié avec succès pour que l’authentification puisse se poursuivre. Cependant, si la vérification d’un module `requisite` échoue, l’utilisateur en est averti immédiatement par le biais d’un message lui indiquant l’échec du premier module `required` *ou* `requisite`.
- `sufficient` — en cas d’échec, les vérifications de modules sont ignorées. Toutefois, si la vérification d’un module portant l’indication `sufficient` est réussie *et* qu’aucun module précédent portant l’indicateur `required` n’a échoué, aucun autre module de ce type n’est nécessaire et l’utilisateur sera authentifié auprès du service.
- `optional` — en cas d’échec, les vérifications de modules sont ignorées. En revanche, si la vérification des modules est réussie, le résultat ne joue aucun rôle dans la réussite ou l’échec global de l’interface de ce module. Un module portant l’indication `optional` devient nécessaire pour la réussite d’une authentification lorsqu’aucun autre module ne fait référence à cette interface. Dans ce cas précis, un module `optional` détermine l’authentification des PAM générale pour cette interface.



Important

L’ordre dans lequel les modules `required` sont appelés n’est pas primordial. Les modules portant l’indication `sufficient` et `requisite` en revanche, donnent à l’ordre une importance vitale.

Il existe désormais pour PAM une nouvelle syntaxe d’indicateurs de contrôle offrant un contrôle encore plus précis. Veuillez lire les documents PAM figurant dans le répertoire `/usr/share/doc/pam-`<*numéro-de-version*>`/` pour obtenir des informations sur cette nouvelle syntaxe (où <*numéro-de-version*> correspond au numéro de version de PAM).

14.3.3. Chemins d’accès aux modules

Les chemins d’accès aux modules indiquent à PAM où trouver les modules enfichables à utiliser avec l’interface de module spécifiée. Normalement, le chemin d’accès complet du module est indiqué, comme par exemple, `/lib/security/pam_stack.so`. Cependant, si ce n’est pas le cas, les système

suppose que le module spécifié se trouve dans le répertoire `/lib/security/`, l'emplacement par défaut des modules PAM.

14.3.4. Arguments des modules

PAM utilise des arguments pour transmettre des informations à un module enfichable lors du processus d'authentification de certains modules.

Par exemple, le module `pam_userdb.so` utilise des indications secrètes stockées dans un fichier de la base de données Berkeley pour authentifier les utilisateurs. La base de données Berkeley est une base de données Open Source intégrée dans de nombreuses applications. Le module nécessite un argument `db` pour spécifier à la base de données Berkeley quelle base de données précise doit être utilisée pour le service demandé.

Une ligne `pam_userdb.so` typique d'un fichier de configuration PAM ressemble à l'extrait suivant:

```
auth      required  /lib/security/pam_userdb.so db=<chemin-au-fichier>
```

Dans l'exemple précédent, remplacez <*chemin-au-fichier*> par le chemin d'accès complet au fichier de la base de données Berkeley DB.

Les arguments non-valides ne sont pas pris en compte et n'ont aucune incidence sur la réussite ou l'échec du module PAM. Toutefois, la plupart des modules rapporteront une erreur dans le fichier `/var/log/messages`.

14.4. Exemples de fichiers de configuration PAM

Ci-dessous figure un exemple de fichier de configuration PAM:

```
#%PAM-1.0
auth      required  /lib/security/pam_securetty.so
auth      required  /lib/security/pam_unix.so shadow nullok
auth      required  /lib/security/pam_nologin.so
account   required  /lib/security/pam_unix.so
password  required  /lib/security/pam_cracklib.so retry=3
password  required  /lib/security/pam_unix.so shadow nullok use_authtok
session   required  /lib/security/pam_unix.so
```

La première ligne est un commentaire, comme l'indique le caractère dièse (#) placé au début de la ligne.

Les lignes deux à quatre empilent trois modules à utiliser pour l'authentification de connexion.

```
auth      required  /lib/security/pam_securetty.so
```

Ce module sert à s'assurer que, *si* l'utilisateur essaie de se connecter en tant que super-utilisateur (ou root), le terminal tty sur lequel il se connecte fait bien partie de la liste se trouvant dans le fichier `/etc/securetty`, *si* ce fichier existe.

```
auth      required  /lib/security/pam_unix.so shadow nullok
```

Ce module invite l'utilisateur à fournir un mot de passe, puis le vérifie à l'aide des informations stockées dans `/etc/passwd` et vérifie s'il existe dans `/etc/shadow`. Le module `pam_unix.so` détecte et utilise automatiquement les mots de passe masqués pour authentifier les utilisateurs. Reportez-vous à la Section 6.5 pour plus d'informations sur les mots de passe masqués.

L'argument `nullok` donne l'instruction au module `pam_unix.so` d'autoriser un mot de passe vide.

```
auth      required  /lib/security/pam_nologin.so
```

Il s'agit de la dernière phase du processus d'authentification. Elle vérifie l'existence du fichier `/etc/nologin`. Si `nologin` n'existe pas et que l'utilisateur n'est pas un super-utilisateur (ou root), l'authentification échoue.



Remarque

Dans cet exemple, les trois modules `auth` sont vérifiés, même si le premier module `auth` échoue. De cette façon, l'utilisateur ne peut pas savoir à quel moment l'authentification a échoué. Si des agresseurs venaient à connaître ces informations, ils pourrait plus facilement déduire de quelle façon pénétrer dans le système.

```
account   required  /lib/security/pam_unix.so
```

Ce module effectuer toute vérification de compte lorsque cela est nécessaire. Par exemple, si des mots de passe masqués ont été activés, l'élément compte du module `pam_unix.so` vérifiera si le compte a expiré ou si l'utilisateur a changé son mot de passe pendant le délai de grâce alloué.

```
password  required  /lib/security/pam_cracklib.so retry=3
```

Si un mot de passe n'est plus valable, l'élément mot de passe du module `pam_cracklib.so` invite l'utilisateur à en fournir un nouveau. Il vérifie ensuite le mot de passe créé afin de déterminer s'il peut être facilement retrouvé par un programme de craquage de mots de passe basé sur des dictionnaires. Si le test du mot de passe échoue, le programme donne à l'utilisateur deux autres possibilités de créer un mot de passe sûr, comme il l'est précisé dans l'argument `retry=3`.

```
password  required  /lib/security/pam_unix.so shadow nullok use_authtok
```

Cette ligne spécifie que, si le programme change le mot de passe de l'utilisateur, il doit le faire en utilisant l'élément `password` du module `pam_unix.so`. Ceci se produit uniquement si la partie `auth` du module `pam_unix.so` détermine que le mot de passe doit être changé.

L'argument `shadow` donne l'instruction au module de créer des mots de passe masqués lors de la mise à jour du mot de passe d'un utilisateur.

L'argument `nullok` donne l'instruction au module d'autoriser l'utilisateur à changer son mot de passe *à partir d'un* mot de passe vide; sinon, un mot de passe non-valide est traité comme un verrouillage de compte.

Le dernier argument de cette ligne, `use_authtok`, est un exemple illustrant bien l'importance de l'ordre lors de l'empilage de modules PAM. Cet argument indique au module de ne pas demander à l'utilisateur un nouveau mot de passe. Au lieu de cela, il accepte tous les mots de passe qui ayant été enregistrés dans le précédent module de mots de passe. De cette façon, tous les nouveaux mots de passe doivent passer le test de sécurité `pam_cracklib.so` avant d'être acceptés.

```
session required /lib/security/pam_unix.so
```

La dernière ligne spécifie que l'élément session du module `pam_unix.so` gérera la session. Ce module enregistre dans `/var/log/messages` le nom d'utilisateur ainsi que le type de service au début et à la fin de chaque session. Il peut être complété en l'empilant avec d'autres modules de session si vous désirez obtenir une fonctionnalité supplémentaire.

L'exemple de fichier de configuration ci-dessous illustre l'empilage du module `auth` pour le programme `rlogin`.

```
#%PAM-1.0
```

```
auth      required    /lib/security/pam_nologin.so
auth      required    /lib/security/pam_securetty.so
auth      required    /lib/security/pam_env.so
auth      sufficient  /lib/security/pam_rhosts_auth.so
auth      required    /lib/security/pam_stack.so service=system-auth
```

Tout d’abord, pam_nologin.so vérifie l’existence de /etc/nologin. S’il existe, seul le super-utilisateur (ou root) se voit autoriser la connexion.

```
auth      required    /lib/security/pam_securetty.so
```

Le module pam_securetty.so empêche les connexions en tant que super-utilisateur sur des terminaux non-sécurisés. Ce faisant, toute tentative d’accès au module rlogin est rejetée en raison des précautions de sécurité.



Astuce

Pour établir une connexion en tant que super-utilisateur, utilisez OpenSSH à la place. Pour plus d’informations sur le protocole SSH, consultez le Chapitre 18.

```
auth      required    /lib/security/pam_env.so
```

Cette ligne charge le module pam_env.so, qui définit les variables d’environnement spécifiées dans /etc/security/pam_env.conf.

```
auth      sufficient  /lib/security/pam_rhosts_auth.so
```

Le module pam_rhosts_auth.so authentifie ensuite l’utilisateur à l’aide de .rhosts dans le répertoire personnel de l’utilisateur. En cas de réussite, PAM authentifie immédiatement la session. En revanche, si pam_rhosts_auth.so échoue lors de l’authentification de l’utilisateur, cette tentative non-réussie n’est pas prise en compte.

```
auth      required    /lib/security/pam_stack.so service=system-auth
```

Si le module pam_rhosts_auth.so ne réussit pas à authentifier l’utilisateur, le module pam_stack.so exécute une authentification normale avec mot de passe.

L’argument service=system-auth indique que l’utilisateur doit passer à travers la configuration PAM pour l’authentification système qui se trouve dans /etc/pam.d/system-auth.



Astuce

Pour éviter que PAM n’invite l’utilisateur à fournir un mot de passe lorsque la vérification securetty échoue, changez l’indicateur du module pam_securetty.so de required à requisite.

14.5. Création des modules PAM

Il est possible à tout moment, d’ajouter des modules d’authentification enfichables pouvant être ensuite utilisés par des applications prenant en charge les PAM. Par exemple, si un développeur élabore une méthode de création de mot de passe unique et écrit un module PAM pour la prendre en charge, les programmes prenant en charge les PAM pourront immédiatement utiliser ce nouveau module ainsi que cette méthode de mot de passe sans avoir à être recompilés ou modifiés. Ainsi, des développeurs et

administrateurs système peuvent combiner et tester rapidement des méthodes d'authentification pour différents programmes sans devoir les recompiler.

La documentation sur l'écriture de modules est fournie avec le système dans le répertoire `/usr/share/doc/pam-<numéro-de-version>/` (où `<numéro-de-version>` correspond au numéro de version de PAM).

14.6. Propriété de PAM et des périphériques

RHL; donne au premier utilisateur privilégié à s'être connecté à la console de la machine la possibilité de manipuler les périphériques et d'exécuter des tâches qui sont normalement réservées au super-utilisateur. Ceci est contrôlé par un module PAM appelé `pam_console.so`.

14.6.1. Propriété des périphériques

Lorsqu'un utilisateur se connecte à une machine utilisant Red Hat Linux, le module `pam_console.so` est appelé par `login` ou par les programmes de connexion graphique **gdm** et **kdm**. Si l'utilisateur est le premier à se connecter à la console physique — que l'on appelle alors *utilisateur console* — le module lui attribue la propriété de périphériques qui appartiennent normalement au super-utilisateur. L'utilisateur console demeure propriétaire de ces périphériques jusqu'à la fin de la dernière session locale de cet utilisateur. Une fois que l'utilisateur s'est déconnecté, la propriété de ces périphériques retourne au super-utilisateur.

Les périphériques affectés incluent notamment les cartes son ainsi que les lecteurs de disquettes et de CD-ROM.

Ainsi, un utilisateur local peut gérer ces périphériques sans être connecté en tant que super-utilisateur, ce qui simplifie les tâches communes de l'utilisateur console.

En modifiant le fichier `/etc/security/console.perms`, l'administrateur peut changer la liste des périphériques contrôlés par `pam_console.so`.

14.6.2. Accès aux applications

L'utilisateur console peut également accéder à n'importe quel programme à l'aide d'un fichier portant le nom de la commande dans le répertoire `/etc/security/console.apps/`.

Un groupe d'applications auquel l'utilisateur console a accès contient trois programmes qui arrêtent ou redémarrent le système, à savoir:

- `/sbin/halt`
- `/sbin/reboot`
- `/sbin/poweroff`

Puisqu'il s'agit d'applications prenant en charge les PAM, le fichier `pam_console.so` est indispensable pour qu'elles puissent fonctionner.

Pour plus d'informations, consultez les pages de manuel relatives à `pam_console`, `console.perms`, `console.apps` et `userhelper`.

14.7. Ressources supplémentaires

Ci-dessous figure une liste de sources d'informations se rapportant à l'utilisation et à la configuration des PAM. Outre ces ressources, consultez également les fichiers de configuration PAM de votre système afin de mieux comprendre leur structure.

14.7.1. Documentation installée

- `man pam` — une bonne introduction à PAM, couvrant la structure ainsi que l'objectif des fichiers de configuration PAM.
- `/usr/share/doc/pam-`<*numéro-de-version*> — contient un guide pour les administrateurs système, *System Administrators' Guide*, un manuel pour les concepteurs de modules, *Module Writers' Manual* et le manuel pour les développeurs d'applications, *Application Developers' Manual*. Il contient également une copie de DCE-RFC 86.0, la norme PAM.

14.7.2. Sites Web utiles

- http://www.kernel.org/pub/linux/libs/pam/ — le site Web de distribution principal pour le projet Linux-PAM contenant des informations sur différents modules PAM, un Forum Aux Questions (FAQ) ainsi que de la documentation supplémentaire sur PAM.



Chapitre 15. Les enveloppeurs TCP et `xinetd`

Le contrôle de l'accès aux services du réseau est l'une des tâches de sécurité les plus importantes à laquelle un administrateur de serveur doit faire face. Heureusement, sous Red Hat Linux il existe un certain nombre d'outils conçus pour cette tâche. Par exemple, le pare-feu basé sur `iptables` filtre les paquets réseau indésirables à partir de la pile réseau du noyau. Pour les services de réseau qui l'utilisent, des *enveloppeurs TCP* ajoutent une couche de protection supplémentaire en déterminant les hôtes autorisés ou non à se connecter à des services de réseau "*enveloppés*". Parmi ces services de réseau enveloppés figure le *super-serveur* `xinetd`. Ce service est baptisé super-serveur parce qu'il contrôle les connexions à un sous-réseau de services et raffine encore plus, le contrôle de l'accès.

La Figure 15-1 est une illustration élémentaire de la manière dont ces outils fonctionnent de concert pour protéger des services de réseau.

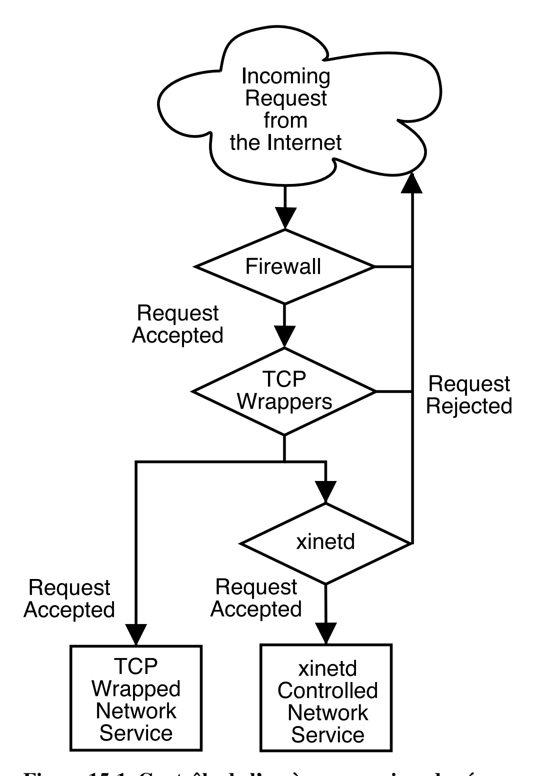


Figure 15-1. Contrôle de l'accès aux services de réseau

Ce chapitre examine d'une part, le rôle des enveloppeurs TCP et de `xinetd` dans le processus de contrôle de l'accès aux services du réseau et, d'autre part, analyse la manière dont ces outils peuvent être utilisés afin d'améliorer aussi bien la gestion des connexions que celle de l'utilisation du système. Pour des informations sur la création de pare-feu avec `iptables`, reportez-vous au Chapitre 16.

15.1. Les enveloppeurs TCP

Le paquetage des enveloppeurs TCP (`tcp_wrappers`) est installé par défaut sous Red Hat Linux et fournit un contrôle de l'accès aux services du réseau, basé sur l'hôte. La bibliothèque `/usr/lib/libwrap.a` représente l'élément le plus important du paquetage. D'une manière générale, un service enveloppé avec TCP est un service qui a été compilé avec la bibliothèque `libwrap.a`.

Lorsqu'une tentative de connexion à un service enveloppé avec TCP est effectuée, le service cherche d'abord les fichiers *d'accès des hôtes (hosts access)* (`/etc/hosts.allow` et `/etc/hosts.deny`) afin de déterminer si l'hôte client est autorisé ou non à se connecter. Il utilise ensuite le démon syslog (`syslogd`) pour écrire le nom de l'hôte envoyant la requête et le nom du service demandé dans `/var/log/secure` ou `/var/log/messages`.

Si un hôte client a la permission de se connecter, les enveloppeurs TCP cèdent le contrôle de la connexion au service demandé et n'interfèrent plus entre l'hôte client et le serveur dans le processus de communication .

Outre le contrôle d'accès et la connexion, les enveloppeurs TCP peuvent activer des commandes afin d'interagir avec le client avant de de refuser ou de céder le contrôle de la connexion au service de réseau demandé.

Étant donne que les enveloppeurs TCP représentent une précieuse adjonction à la panoplie des outils de sécurité de tout administrateur de serveur, la plupart des services de réseau sous Red Hat Linux sont étroitement liés à la bibliothèque `libwrap.a`. Parmi ces applications figurent `/usr/sbin/sshd`, `/usr/sbin/sendmail` et `/usr/sbin/xinetd`.



Remarque

Afin de déterminer si un binaire de service de réseau est lié à `libwrap.a`, tapez la commande suivante en étant connecté en tant que super-utilisateur (ou root):

```
strings -f <nom-binaire> | grep hosts_access
```

en remplaçant bien <*nom-binaire*> par le nom du binaire du service de réseau.

15.1.1. Avantages des enveloppeurs TCP

Les enveloppeurs TCP offrent deux avantages de base par rapport à d'autres techniques de contrôle de services de réseau:

- *La transparence des opérations aussi bien pour l'hôte client que pour le service de réseau enveloppé.* — Ni le client établissant la connexion, ni le service de réseau enveloppé ne remarqueront que des enveloppeurs TCP sont utilisés. Les utilisateurs légitimes sont connecté et branchés au service demandé alors que les connexions provenant de clients non-authorisés sont refusées.
- *Une gestion centrale de protocoles multiples.* — étant donné que les enveloppeurs TCP fonctionnent indépendemment des services de réseau qu'ils protègent, ils permettent à de nombreuses applications serveurs de partager un jeu de fichiers de configuration commun offrant ainsi une gestion simplifiée.

15.2. Fichiers de configuration des enveloppeurs TCP

Afin de déterminer si un ordinateur client est autorisé à se connecter à un service, les enveloppeurs TCP référencent les deux fichiers suivants, couramment appelés fichiers d'accès des hôtes:

- `/etc/hosts.allow`
- `/etc/hosts.deny`

Lorsqu'une requête cliente est reçue par un service enveloppé avec TCP, ce dernier suit les étapes élémentaires suivantes:

1. *Le service référence `/etc/hosts.allow`.* — Le service enveloppé avec TCP fait l'analyse grammaticale du fichier `/etc/hosts.allow` de manière séquentielle et applique la première règle spécifiée pour ce service. Si une règle correspond au service, il autorise la connection. Sinon, il passe à la deuxième étape.
2. *Le service référence `/etc/hosts.deny`.* — Le service enveloppé avec TCP fait l'analyse grammaticale du fichier `/etc/hosts.deny` de manière séquentielle. Si une règle correspond au service, il refuse la connection. Sinon, il autorise l'accès au service.

Ci-après figurent des points importants à prendre en compte lors de l'utilisation d'enveloppeurs TCP pour protéger des services de réseau:

- Parce que les règles d'accès contenues dans le fichier `hosts.allow` sont appliquées en premier, elles ont priorité par rapport aux règles spécifiées dans le fichier `hosts.deny`. Par conséquent, si l'accès à un service est autorisé dans `hosts.allow`, mais qu'une règle refusant l'accès à ce même service est contenue dans le fichier `hosts.deny`, cette dernière ne sera pas prise en compte.
- Étant donné que les règles dans chaque fichier sont lues de haut en bas et que la première règle appliquée à un service donné est la seule règle prise en compte, l'ordre de ces dernières est essentiel.
- Si aucune règle contenue dans l'un ou l'autre des fichiers ne s'appliquent au service, ou si aucun de ces fichiers n'existe, l'accès au service est autorisé.
- Des services enveloppés avec TCP ne mettent pas en cache les règles des fichiers d'accès d'hôtes, ainsi, tout changement apporté à `hosts.allow` ou `hosts.deny` prend effet immédiatement sans devoir redémarrer les services de réseau.

15.2.1. Formatage des règles d'accès

Le format est le même pour le fichier `/etc/hosts.allow` et le fichier `/etc/hosts.deny`. Toute ligne vierge ou commençant pas un symbole dièse (`#`) n'est pas prise en compte; de plus, chaque règle doit figurer sur sa propre ligne.

Chaque règle utilise le format élémentaire suivant pour contrôler l'accès aux services de réseau:

```
<daemon list>: <client list> [: <option>: <option>: ...]
```

- *<daemon list>* — correspond à une liste de noms de processus (*pas* des noms de services) ou *caractère générique* (ou 'wildcard') `ALL`, séparés par des virgules (Consultez la Section 15.2.1.1). La liste des démons accepte aussi les *opérateurs* énumérés dans la Section 15.2.1.3 afin d'offrir une plus grande flexibilité.
- *<client list>* — correspond à une liste de noms d'hôtes, d'adresses IP hôtes, de *gabarits* spéciaux, (voir la Section 15.2.1.2) ou de jokers ('wildcards') (voir la Section 15.2.1.1), séparés par des virgules, identifiant les hôtes auxquels la règle s'applique. La liste de clients accepte également les opérateurs énumérés dans la Section 15.2.1.3 afin d'offrir une plus grande flexibilité.

- <*option*> — correspond à une action facultative ou à une liste d'actions facultatives séparées par des virgules, devant être exécutées lorsque la règle est appliquée. Les champs d'options prennent en charge les *expansions* (voir la Section 15.2.3.4) et peuvent être utilisés pour lancer des commandes du shell, autoriser ou refuser l'accès et modifier le comportement de connexion (voir la Section 15.2.3).

Ci-après figure un exemple élémentaire de règle d'accès d'hôte:

```
vsftpd : .example.com
```

Cette règle donne aux enveloppeurs TCP l'instruction de surveiller les connexions établies au démon FTP (`vsftpd`) à partir de tout hôte du domaine `example.com`. Si cette règle apparaît dans `hosts.allow`, la connexion sera acceptée. En revanche, si la règle est présente dans `hosts.deny`, la connexion sera refusée.

La règle d'accès d'hôtes suivante est plus complexe et inclus deux champs d'option:

```
sshd : .example.com  \
: spawn /bin/echo `/bin/date` access denied>>/var/log/sshd.log \
: deny
```

Notez que dans cet exemple, chaque champ d'option est précédé de la barre oblique inverse (\). L'utilisation de ce symbole empêche que la règle n'échoue en raison de sa longueur.



Avertissement

Si la dernière ligne du fichier d'accès d'hôtes ne correspond pas au caractère symbolisant une nouvelle ligne (créé en pressant sur la touche [Entrée]), la dernière règle du fichier échouera et un message d'erreur sera journalisé soit dans `/var/log/messages`, soit dans `/var/log/secure`. Ceci s'applique aussi à des lignes de règles qui s'étendent sur plusieurs lignes sans inclure le symbole de la barre oblique inverse. L'exemple suivant illustre la partie pertinente d'un message de journalisation faisant référence à l'échec d'une règle en raison de l'une ou l'autre des circonstances mentionnées ci-dessus:

```
warning: /etc/hosts.allow, line 20: missing newline or line too long
```

Cette exemple de règle stipule que si un hôte d domaine `example.com` essaie d'établir une connexion au démon SSH (`sshd`), la commande `echo` doit être exécutée (permettant de journaliser cette tentative de connexion dans un fichier spécial) et la connexion refusée. Puisque la directive optionnelle `deny` est utilisée, cette ligne entraînera un refus de l'accès même si elle figure dans le fichier `hosts.allow`. Pour des informations plus détaillées sur les options disponibles, reportez-vous à la Section 15.2.3.

15.2.1.1. Jokers (ou 'Wildcards')

Les jockers permettent aux enveloppeurs TCP d'autoriser plus facilement les groupes de démons et les hôtes. Ils sont le plus souvent utilisés dans le champ de la liste de clients des règles d'accès.

Les jokers suivants peuvent être utilisés:

- `ALL` — Accorde à tout client l'accès d'un service. Ce joker peut être utilisé aussi bien pour la liste des démons que celle des clients.
- `LOCAL` — Autorise tout hôte ne contenant pas de point (`.`), comme par exemple un hôte local.
- `KNOWN` — Autorise tout hôte dont le nom ou l'adresse d'hôte sont connus ou lorsque l'utilisateur est connu.

- `UNKNOWN` — Autorise tout hôte dont le nom ou l'adresse d'hôte sont inconnus ou lorsque l'utilisateur est inconnu.
- `PARANOID` — Autorise tout hôte dont le nom d'hôte ne correspond pas à l'adresse d'hôte.



Attention

Les jokers `KNOWN`, `UNKNOWN` et `PARANOID` doivent être utilisés avec précaution, car une rupture de la résolution de noms peut empêcher des utilisateurs légitimes d'accéder au service.

15.2.1.2. Patterns

Les gabarits peuvent être utilisés dans le champ de la liste de clients des règles d'accès afin de spécifier de manière plus précise des groupes d'hôtes clients.

Ci-dessous figure une liste des gabarits les plus communément acceptés pour une entrée dans la liste de clients:

- *Nom d'hôte commençant par un point (.)* — En plaçant un point au début d'un nom d'hôte, tous les hôtes partageant l'élément listé du nom seront autorisés. L'exemple suivant s'appliquerait à tout hôte du domaine `example.com`:
  ```
  ALL : .example.com
  ```
- *Adresse IP finissant par un point (.)* — En plaçant un point à la fin d'une adresse IP, tous les hôtes partageant les premiers groupes numériques d'une adresse IP seront autorisés. L'exemple suivant s'appliquerait à tout hôte du réseau `192.168.x.x`:
  ```
  ALL : 192.168.
  ```
- *Paire adresse IP/masque réseau* — Les expression de masques réseau peuvent également être utilisées comme un gabarit pour contrôler l'accès à un groupe particulier d'adresses IP. L'exemple suivant s'appliquerait à tout hôte doté d'une adresse IP comprise entre `192.168.0.0` et `192.168.1.255`:
  ```
  ALL : 192.168.0.0/255.255.254.0
  ```
- *L'astérisque (*)* — Des astérisques peuvent être utilisés pour autoriser des groupes entiers de noms d'hôtes ou d'adresses IP, à condition qu'ils ne fassent pas aussi partie d'une liste de clients contenant d'autres types de gabarits. L'exemple suivant s'appliquerait à tout hôte du domaine `example.com`:
  ```
  ALL : *.example.com
  ```
- *La barre oblique (/)* — Si une liste de clients commence par une barre oblique, elle est considérée comme un nom de fichier. Ce symbole est utile lorsque des règles spécifiant de nombreux hôtes sont nécessaires. L'exemple suivant renvoie les enveloppeurs TCP au fichier `/etc/telnet.hosts` pour toutes les connexion à Telnet:
  ```
  in.telnetd : /etc/telnet.hosts
  ```

D'autres gabarits, moins utilisés sont également acceptés par les enveloppeurs TCP. Consultez la section 5 de la page de manuel relative à l'accès d'hôtes (hosts_access) pour de plus amples informations.



Avertissement

Soyez très prudent lorsque vous créez des règles nécessitant une résolution de nom, comme par exemple, noms d'hôtes et noms de domaines. Des agresseurs peuvent recourir à une variétés de tactiques pour contourner une résolution de nom précise. En outre, toute perturbation du service DNS empêcherait même des utilisateurs autorisés d'utiliser les services du réseau.

Il est préférable, autant que possible, d'utiliser des adresses IP.

15.2.1.3. Opérateurs

À l'heure actuelle, les règles de contrôle d'accès acceptent un opérateur, à savoir `EXCEPT`. Il peut être utilisé aussi bien dans la liste des démons d'une règle que dans celle des clients.

L'opérateur `EXCEPT` permet d'introduire des exceptions spécifiques à des correspondances générales au sein de la même règle.

Dans l'exemple ci-dessous tiré d'un fichier `hosts.allow`, tous les hôtes `example.com` sont autorisés à se connecter aux services sauf `cracker.example.com`:

```
ALL: .example.com EXCEPT cracker.example.com
```

Dans l'autre exemple ci-dessous tiré du fichier `hosts.allow`, les clients du réseau 192.168.0.*x* peuvent utiliser tous les services sauf FTP:

```
ALL EXCEPT vsftpd: 192.168.0.
```



Remarque

D'un point de vue organisationnel, il est souvent plus facile d'utiliser les opérateurs `EXCEPT` avec parcimonie, en choisissant plutôt de placer les exceptions à la règle dans l'autre fichier de contrôle d'accès. Ce faisant, d'autres administrateurs peuvent examiner rapidement le fichier approprié pour voir quels hôtes doivent être autorisés ou refusés pour quels services, sans devoir trier les divers opérateurs `EXCEPT`.

15.2.2. Portmap et les enveloppeurs TCP

Lors de la création de règles de contrôle d'accès pour `portmap`, n'utilisez pas les noms d'hôtes car son implémentation des enveloppeurs TCP ne prend pas en charge la consultation des hôtes. Pour cette raison, utilisez seulement des adresses IP ou le mot-clé `ALL` lors de la spécification des hôtes dans `hosts.allow` ou `hosts.deny`.

De plus, les changements apportés aux règles de contrôle d'accès `portmap` ne prennent pas toujours effet immédiatement.

Étant donné que des services très populaires comme NIS et NFS, dépendent de `portmap` pour leur fonctionnement, assurez-vous de bien prendre ces limitations en compte.

15.2.3. Les champs d'options

Au delà de la simple autorisation ou du refus d'accès, l'implémentation Red Hat Linux des enveloppeurs TCP prend en charge des extensions au langage utilisé pour le contrôle d'accès au moyen des champs d'options. En utilisant des champs d'options au sein des règles d'accès d'hôtes, les administrateurs peuvent accomplir un vaste éventail de tâches, comme entre autres, la modification du comportement de journalisation, la consolidation du contrôle d'accès et le lancement de commandes du shell.

15.2.3.1. Journalisation

Les champs d'options permettent aux administrateurs de changer facilement la fonction de journalisation et le niveau de gravité d'une règle à l'aide de la directive `severity`.

Dans l'exemple suivant, les connexions au démon SSH à partir de tout hôte du domaine `example.com` sont journalisées dans le journal `authpriv` par défaut (car aucune valeur de fonction n'est spécifiée) avec une priorité `emerg`:

```
sshd : .example.com : severity emerg
```

Il est également possible de spécifier un service à l'aide de l'option `severity`. L'exemple suivant journalise tous les hôtes du domaine `example.com` essayant de se connecter au service SSH dans `local0` avec la priorité `alert`:

```
sshd : .example.com : severity local0.alert
```



Remarque

Dans la pratique, cet exemple ne fonctionnera pas tant que le démon syslog (`syslogd`) est configuré pour qu'il journalise `local0`. Consultez les pages de manuel relatives à `syslog.conf` pour de plus amples informations sur la configuration personnalisée des fonctions de journalisation.

15.2.3.2. Contrôle d'accès

Les champs d'options permettent également aux administrateurs d'autoriser ou de refuser de manière explicite des hôtes dans une seule règle en ajoutant la directive `allow` ou `deny` en tant que dernière option.

Par exemple, les deux règles suivantes permettent des connexions SSH à partir de `client-1.example.com`, mais les refusent à partir de `client-2.example.com`:

```
sshd : client-1.example.com : allow
sshd : client-2.example.com : deny
```

En permettant le contrôle d'accès sur la base de règles individuelles, le champs d'options parmet aux administrateurs de consolider toutes les règles d'accès dans un seul et même fichier: soit `hosts.allow`, soit `hosts.deny`. Pour certains, cette méthode est la manière la plus simple d'organiser des règles d'accès.

15.2.3.3. Commandes du Shell

Les champs d'options permettent aux règles d'accès de lancer des commandes du shell au moyen des deux directives suivantes:

- `spawn` — Lance une commande du shell en tant que processus enfant. Cette directive permet d'effectuer des tâches comme l'utilisation de `/usr/sbin/safe_finger` pour obtenir des informations supplémentaires sur le client faisant une requête ou pour créer des fichiers de journalisation spéciaux en utilisant la commande `echo`.

  Dans l'exemple suivant, les clients essayant d'accéder aux services Telnet à partir du domaine `example.com` sont journalisés dans un fichier spécial:

  ```
  in.telnetd : .example.com \
  ```

```
    : spawn /bin/echo `/bin/date` from %h>>/var/log/telnet.log \
    : allow
```

- `twist` — Remplace le services demandé par la commande spécifiée. Cette directive est souvent utilisée pour créer des pièges pour les agresseurs. Elle peut également être utilisée pour envoyer des messages à des clients se connectant. La commande twist doit se trouver à la fin de la ligne de règles.

  Dans l'exemple suivant, les clients essayant d'accéder aux services FTP à partir du domaine `example.com` reçoivent un message envoyé au moyen de la commande `echo`:

  ```
  vsftpd : .example.com \
  : twist /bin/echo "421 Bad hacker, go away!"
  ```

Pour de plus amples informations sur les options des commandes du shell, consultez la page de manuel relative à `hosts_options`.

15.2.3.4. Expansions

Les expansions, lorsqu'elles sont utilisées de concert avec les directives `spawn` et `twist` permettent d'obtenir des informations sur le client, le serveur et les processus impliqués.

Ci-après figure une liste des expansions prises en charge:

- `%a` — L'adresse IP du client.
- `%A` — L'adresse IP du serveur.
- `%c` — Fournit diverses informations sur le client, comme les noms d'utilisateur et d'hôte, ou le nom d'utilisateur et l'adresse IP.
- `%d` — Le nom du processus du démon.
- `%h` — Le nom d'hôte du client (ou adresse IP, si le nom d'hôte n'est pas disponible).
- `%H` — Le nom d'hôte du serveur (ou adresse IP, si le nom n'est pas disponible).
- `%n` — Le nom d'hôte du client. S'il n'est pas disponible, c'est `unknown` qui est imprimé. Si les noms d'hôte et d'adresse du client ne correspondent pas, c'est `paranoid` qui est imprimé.
- `%N` — Le nom d'hôte du serveur. Si celui-ci n'est pas disponible, c'est `unknown` qui est imprimé. Si les noms d'hôte et d'adresse du client ne correspondent pas, c'est `paranoid` qui est imprimé.
- `%p` — L'ID du processus de démon.
- `%s` — Diverses types d'informations sur le serveur, comme le processus de démon ou l'hôte ou l'adresse IP du serveur.
- `%u` — Le nom d'utilisateur du client. Si celui-ci n'est pas disponible, c'est `unknown` qui est imprimé.

L'exemple de règle suivant utilise une expansion en même temps que la commande `spawn` pour identifier l'hôte client dans un fichier de journalisation personnalisé.

Elle indique aux envelopppeurs TCP que, lors de toute tentative de connexion au démon SSH (`sshd`) à partir d'un hôte du domaine `example.com`, ils doivent exécuter la commande `echo` afin de journaliser non seulement la tentative, mais également le nom d'hôte du client (à l'aide de l'expansion `%h`), dans un fichier spécial:

```
sshd : .example.com  \
: spawn /bin/echo `/bin/date` access denied to %h>>/var/log/sshd.log \
: deny
```

De même, des expansions peuvent être utilisées pour personnaliser les messages renvoyés au client. Dans l'exemple suivant, les clients essayant de se connecter aux services FTP à partir du domaine `example.com` sont informés qu'ils ont été bannis du serveur:

```
vsftpd : .example.com \
: twist /bin/echo "421 %h has been banned from this server!"
```

Pour une explication complètes des expansions disponibles et des options supplémentaires de contrôle d'accès, reportez-vous à la section 5 de la page de manuel relative à `hosts_access` (`man 5 hosts_access`) et à la page de manuel relative à `hosts_options`.

Pour des ressources supplémentaires sur les enveloppeurs TCP, reportez-vous à la Section 15.5.

15.3. xinetd

Le démon `xinetd` est un *super-service* enveloppé par TCP permettant de contrôler l'accès à un sous-réseau de services de réseau populaires parmi lesquels figurent FTP, IMAP et Telnet. Il permet également de spécifier des options de configuration spécifiques aux services en matière de contôle d'accès, journalisation améliorée, liaison, redirection et de contrôle d'utilisation de ressources.

Lorsqu'un hôte client essaie de se connecter à un service du réseau contrôlé par `xinetd`, le super-service reçoit la requête et vérifie l'existence de toute règle de contrôle d'accès des enveloppeurs TCP. Si l'accès est autorisé, `xinetd` vérifie non seulement que la connexion est également bien autorisée selon ses propres règles d'accès pour ce service mais que le service n'utilise pas plus de ressources que la quantité qui lui est attribuée et qu'il ne commet aucune d'infraction aux règles définies. Il démarre alors une instance du service demandé et lui cède le contrôle de la connexion. Une fois la connexion établie, `xinetd` n'interfère plus dans le processus de communication entre l'hôte client et le serveur.

15.4. Fichiers de configuration de xinetd

Les fichiers de configuration de `xinetd` sont les suivants:

- le fichier `/etc/xinetd.conf` — le fichier de configuration général de `xinetd` configuration file;
- le répertoire `/etc/xinetd.d/` — le répertoire contenant tous les fichiers spécifiques aux services.

15.4.1. Le fichier /etc/xinetd.conf

Le fichier `/etc/xinetd.conf` contient des paramètres de configuration généraux ayant une influence sur tous les services placés sous le contrôle de `xinetd`. Il n'est lu que lors du lancement du service `xinetd`, par conséquent, afin que des changement apportés à la configuration puissent prendre effet, l'administrateur doit redémarrer le service `xinetd`. Ci-après figure un exemple de fichier `/etc/xinetd.conf`:

```
defaults
{
        instances               = 60
        log_type                = SYSLOG authpriv
        log_on_success          = HOST PID
        log_on_failure          = HOST
        cps                     = 25 30
}
includedir /etc/xinetd.d
```

Ces lignes contrôlent divers aspects de xinetd:

- instances — détermine le nombre maximum de requêtes qu'un service xinetd peut gérer à un moment donné.
- log_type — indique à xinetd d'utiliser le journal authpriv qui enregistre des entrées de journalisation dans le fichier /var/log/secure. En ajoutant ici une directive comme FILE /var/log/xinetdlog un fichier de journalisation personnalisé portant le nom xinetdlog sera créé dans le répertoire /var/log/.
- log_on_success — Configure xinetd de façon à ce qu'il journalise si la connexion est établie avec succès. Par défaut sont enregistrés aussi bien l'adresse IP de l'hôte distant que l'ID de processus du serveur traitant la requête.
- log_on_failure — Configure xinetd de façon à ce qu'il journalise si la connexion échoue ou si elle n'est pas autorisée.
- cps — Configure xinetd de manière à n'autoriser que 25 connexions par seconde à un service donné. Si cette limite est atteinte, le service est retiré pendant 30 secondes.
- includedir /etc/xinetd.d/ — Inclut des options stipulées dans les fichiers de configuration spécifiques aux services qui se trouvent dans le répertoire /etc/xinetd.d/. Reportez-vous à la Section 15.4.2 pour plus d'informations sur ce répertoire.



Remarque

Les paramètres de log_on_success et log_on_failure dans /etc/xinetd.conf sont souvent encore modifiés dans les fichiers de journalisation spécifique à chaque service. Pour cette raison, plus d'informations sont pafois enregistrées pour un service donné que ce qui est en fait spécifié dans le fichier-même. Reportez-vous à la Section 15.4.3.1 pour de plus amples informations sur les options de journalisation.

15.4.2. Le répertoire /etc/xinetd.d/

Le répertoire /etc/xinetd.d/ contient les fichiers de configuration relatifs à chaque service géré par xinetd; ces derniers portent un nom faisant référence au service. De même que pour xinetd.conf, ce fichier est lu seulement lorsque le service xinetd est lancé. Ainsi, afin que tout changement puisse prendre effet, l'administrateur doit relancer le service xinetd.

Le format des fichiers dans le répertoire /etc/xinetd.d/ se base sur les même conventions que /etc/xinetd.conf. La raison essentielle de leur stockage dans des fichiers de configuration séparés est de faciliter la personnalisation et d'éviter qu'elle n'affecte trop les autres services.

Pour comprendre comment ces fichiers sont structurés, examinons le fichier /etc/xinetd.d/telnet:

```
service telnet
{
        flags           = REUSE
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/in.telnetd
        log_on_failure  += USERID
        disable         = yes
}
```

Ces lignes contrôlent différents aspects du service telnet:

- service — Définit le nom du service, généralement pour correspondre à un service énuméré dans le fichier /etc/services.
- flags — Définit tout attribut pour la connexion, parmi la variété disponible. REUSE donne l'instruction à xinetd de réutiliser le support pour une connexion Telnet.
- socket_type — Spécifie le connecteur réseau comme étant de type stream.
- wait — Détermine si le service est mono-fil ('single-threaded', yes) ou multi-fils ('multi-threaded', no).
- user — Détermine l'ID d'utilisateur sous lequel le processus sera exécuté.
- server — Définit le fichier binaire exécutable à lancer.
- log_on_success — Détermine les paramètres de journalisation de log_on_success, en plus de ceux déjà définis dans xinetd.conf.
- log_on_failure — Détermine les paramètres de journalisation de log_on_failure en plus de ceux déjà définis dans xinetd.conf.
- nice — Détermine le niveau de priorité du serveur.
- disable — Détermine si le service est actif ou non.

15.4.3. Modification des fichiers de configuration de xinetd

De nombreuses directives existent pour les services protégés de xinetd. Cette section souligne certaines des options les plus couramment utilisées.

15.4.3.1. Options de journalisation

Les options de journalisation suivantes sont disponibles aussi bien pour /etc/xinetd.conf que pour les fichiers de configuration spécifiques à certains services stockés dans le répertoire /etc/xinetd.d/.

Ci-dessous figure une liste des options de journalisation les plus couramment utilisées:

- ATTEMPT — Enregistre une tentative qui a échoué (log_on_failure).
- DURATION — Enregistre la durée d'utilisation du service par un système distant (log_on_success).
- EXIT — Enregistre le statut de sortie ou le signal de fin d'un service (log_on_success).
- HOST — Enregistre l'adresse IP de l'hôte distant (log_on_failure et log_on_success).
- PID — Enregistre l'ID de processus du serveur recevant la requête (log_on_success).
- RECORD — Enregistre des informations sur le système distant dans le cas où le service ne peut pas être démarré. Seuls les services particuliers, comme login et finger peuvent utiliser cette option (log_on_failure).
- USERID — Enregistre l'utilisateur distant selon la méthode définie dans RFC 1413 pour tous les services en flux continu multi-fils (multi-threaded) (log_on_failure et log_on_success).

Pour une liste complètes des options de journalisation, consultez les pages de manuel relatives à xinetd.conf.

15.4.3.2. Options de contrôle d'accès

Les utilisateurs des services `xinetd` peuvent choisir d'utiliser les règles de contrôle d'accès des enveloppeurs TCP, de fournir le contrôle d'accès par le biais des fichiers de configuration de `xinetd` ou de recourir à un mélange des deux. Des informations sur l'utilisation des fichiers de contrôle d'accès par l'hôte des enveloppeurs TCP se trouvent dans la Section 15.2. Cette section examine l'utilisation de `xinetd` pour contrôler l'accès aux services.



Remarque

À la différence des enveloppeurs TCP, les changements des contrôles d'accès ne prennent effet que si l'administrateur de `xinetd` administrator relance le service `xinetd`.

Le contrôle d'accès des hôtes à `xinetd` est différent de la méthode utilisée par les enveloppeurs TCP. Alors que ces derniers placent toutes les configurations d'accès dans deux fichiers, soit `/etc/hosts.allow` et `/etc/hosts.deny`, le fichier de chaque service dans `/etc/xinetd.d` peut contenir ses propres règles de contrôle d'accès.

Les options suivantes d'accès des hôtes sont prises en charge par `xinetd`:

- `only_from` — Permet seulement aux hôtes spécifiés d'utiliser le service.
- `no_access` — Empêche les hôtes spécifiés d'utiliser le service.
- `access_times` — Spécifie la fourchette de temps pendant laquelle un service particulier peut être utilisé. Cette durée doit être stipulée dans une notation sur 24 heures et selon le format `HH:MM-HH:MM`.

Les options `only_from` et `no_access` peuvent utiliser une liste d'adresses IP ou noms d'hôte, ou peuvent spécifier un réseau entier. Comme le font les enveloppeurs TCP, la combinaison du contrôle d'accès `xinetd` avec une configuration de journalisation améliorée permet d'accroître la sécurité non seulement en empêchant les requêtes provenant d'hôtes bannis mais en enregistrant également des informations détaillées sut chaque tentative de connexion.

Par exemple, le fichier suivant `/etc/xinetd.d/telnet` peut être utilisé pour non seulement bloquer l'accès à Telnet partir d'un groupe de réseau spécifique mais également limiter la fourchette de temps générale pendant laquelle même les utilisateurs autorisés peuvent se connecter:

```
service telnet
{
        disable         = no
        flags           = REUSE
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/in.telnetd
        log_on_failure  += USERID
        no_access       = 10.0.1.0/24
        log_on_success  += PID HOST EXIT
        access_times    = 09:45-16:15
}
```

Dans cet exemple, lorsque tout système client provenant du réseau 10.0.1.0/24, tel que 10.0.1.2, essaie d'accéder au service Telnet, il recevra un message au contenu suivant:

```
Connection closed by foreign host.
```

De plus, la tentative de connexion est enregistrée dans `/var/log/secure` de la manière suivante:

```
May 15 17:38:49 boo xinetd[16252]: START: telnet pid=16256 from=10.0.1.2
May 15 17:38:49 boo xinetd[16256]: FAIL: telnet address from=10.0.1.2
May 15 17:38:49 boo xinetd[16252]: EXIT: telnet status=0 pid=16256
```

Lors de l'utilisation des enveloppeurs TCP de concert avec les accès de contrôle `xinetd`, il est important de bien comprendre la relation entre les deux mécanismes de contrôle d'accès.

Les informations suivantes montrent l'ordre des opérations suivi par `xinetd` lorsqu'un client demande à établir une connexion:

1. Le démon `xinetd` accède aux règles d'accès par hôte des enveloppeurs TCP et ce par le biais d'un appel à la bibliothèque `libwrap.a`. Si une règle de refus s'applique à l'hôte client, la connexion est abandonnée. Si une règle d'autorisation s'applique à l'hôte client, la connexion est passée à `xinetd`.
2. Le démon `xinetd` vérifie ses propres règles de contrôle d'accès aussi bien pour le service `xinetd` que pour le service demandé. Si une règle de refus s'applique à l'hôte client, la connexion est abandonnée. Sinon, `xinetd` démarre une instance du service demandé et lui cède le contrôle de la connexion.



Important

Il est important de bien faire attention lors de l'utilisation des contrôles d'accès des enveloppeurs TCP en concert avec les contrôles d'accès de `xinetd`. En effet, une mauvaise configuration peut entraîner des effets indésirables.

15.4.3.3. Options de liaison et redirection

Les fichiers de configuration de service pour `xinetd` prennent en charge la liaison du service à une adresse IP et la redirection de requêtes entrantes pour ce service vers une autre adresse IP, nom d'hôte, ou port.

La liaison est contrôlée par l'option `bind` dans les fichiers de configuration d'un service spécifique et lie le service à une adresse IP dans le système. Une fois configurée, l'option `bind` autorise seulement des requêtes pour l'adresse IP adéquate pour accéder au service. De cette manière, différents services peuvent se trouver liés à différentes interfaces réseau selon les besoins.

Cela est particulièrement utile pour les systèmes à adaptateurs de réseaux multiples ou ayant de multiples adresses IP configurées. Sur un tel système, des services non-sécurisés, comme Telnet, peuvent être configurés de manière à recevoir des requêtes seulement sur l'interface connectée à un réseau privé et pas l'interface connectée à l'Internet.

L'option `redirect` accepte une adresse IP ou nom d'hôte suivi par un numéro de port. Elle permet de configurer le service de manière à ce qu'il redirige toute requête pour ce service vers l'hôte et le numéro de port spécifié. Cette fonction peut être employée pour diriger vers un autre numéro de port sur le même système, rediriger la requête vers une autre adresse IP sur la même machine, rediriger la requête vers un système et numéro de port totalement différents, ou pour toute combinaison de ces options. De cette façon, un utilisateur se connectant à un certain service sur un système peut être rerouté vers un autre système sans interruption.

Le démon `xinetd`peut accomplir cette redirection en produisant un processus qui reste actif pour la durée de la connexion entre l'ordinateur du client effectuant la requête et l'hôte fournissant réellement le service, transférant les données entre les deux systèmes.

Les avantages des options `bind` et `redirect` sont les plus évidents lorsque ces options sont utilisées ensemble. En liant un service à une adresse IP particulière sur un système puis en redirigeant les requêtes pour ce service vers une seconde machine que seule la première peut percevoir, il est possible

d'utiliser un système interne pour fournir des services à un réseau totalement différent. Ces options peuvent également être utilisées pour non seulement limiter l'exposition d'un service particulier sur un ordinateur multi-sites à une adresse IP connue mais aussi pour rediriger toute requête pour ce service vers une autre machine spécialement configurée à cet effet.

Examinons par exemple le cas d'un système utilisé comme pare-feu avec cette configuration pour son service Telnet:

```
service telnet
{
        socket_type  = stream
        wait   = no
        server   = /usr/sbin/in.telnetd
        log_on_success  += DURATION USERID
        log_on_failure  += USERID
        bind                    = 123.123.123.123
        redirect                = 10.0.1.13 21 23
}
```

Les options `bind` et `redirect` dans ce fichier garantissent que le service Telnet sur cette machine est lié à l'adresse IP externe (123.123.123.123), celle qui prend en charge l'Internet. De plus, toute requête de service Telnet envoyée vers 123.123.123.123 est redirigée via un second adaptateur de réseau vers une adresse IP interne (10.0.1.13) à laquelle seuls le pare-feu et les systèmes internes peuvent accéder. Le pare-feu envoie alors la communication entre les deux systèmes, et le système se connectant pense qu'il est connecté à 123.123.123.123 alors qu'en fait il est connecté à une machine différente.

Cette fonction est particulièrement utile pour les utilisateurs avec connexion à large bande et avec seulement une adresse IP fixe. Lors de l'utilisation de la traduction d'adresse de réseau (ou NAT de l'anglais 'Network Address Translation'), les systèmes situés derrière la machine passerelle, qui utilisent des adresses IP exclusivement internes, ne sont pas disponibles depuis l'extérieur du système de passerelle. Toutefois, avec certains services contrôlés par `xinetd` et configurés avec les options `bind` et `redirect`, la machine passerelle peut servir de proxy entre les systèmes externes et une machine interne particulière configurée pour fournir le service en question. De plus, les diverses options de contrôle d'accès `xinetd` et de journalisation peuvent servir à une protection supplémentaire, comme pour limiter le nombre de connexions simultanées pour ce service redirigé.

15.4.3.4. Options de gestion de ressources

Le démon `xinetd` permet d'ajouter un niveau élémentaire de protection contre des attaques de Refus de service (ou DoS, de l'anglais 'Denial of Service'). Ci-dessous figure une liste des directives pouvant aider à limiter l'efficacité de telles attaques:

- `per_source` — Détermine le nombre maximum d'instances d'un service spécifique pour une adresse IP d'origine particulière. Elle n'accepte que comme argument que des chiffres entiers et peut être utilisée aussi bien dans `xinetd.conf` que dans des fichiers de configuration spécifiques à un service stockés dans le répertoire `xinetd.d/`.
- `cps` — Détermine le nombre maximum de connexions par seconde. Cette directive accepte deux arguments avec des valeurs entières, séparés par un espace blanc. Le premier représente le nombre maximum de connexions autorisées à un service par seconde. Le deuxième correspond au nombre de secondes pendant lequel `xinetd` doit attendre avant de réactiver le service. Il n'accepte que des nombres entiers comme argument et peut être utilisé aussi bien dans `xinetd.conf` que dans les fichiers de configuration spécifiques au service du répertoire `xinetd.d/`.
- `max_load` — Définit le seuil d'utilisation d'un processeur (CPU) pour un service. Cette directive accepte un argument avec une valeur flottante.

Il existe encore d'autres options de gestion de ressources utilisables avec `xinetd`. Reportez-vous au chapitre intitulé *Sécurité du serveur* (Server Security) du *Guide de sécurité de Red Hat Linux* pour obtenir de plus amples informations. Consultez également la page de manuel relative à `xinetd.conf`.

15.5. Ressources supplémentaires

Des informations supplémentaires sur les enveloppeurs TCP et `xinetd` sont disponibles aussi bien sur le système lui-même que sur le Web.

15.5.1. Documentation installée

La documentation installée sur votre système est un bon endroit pour commencer des recherches sur les enveloppeurs TCP, sur `xinetd` et sur les options de configuration de contrôle d'accès.

- `/usr/share/doc/tcp_wrappers-`<*version*>`/` — Contient un fichier `README` décrivant le fonctionnement des enveloppeurs TCP et les divers risques potentiels d'usurpation d'adresse et de nom d'hôte.
- `/usr/share/doc/xinetd-`<*version*>`/` — Comprend un fichier `README` qui examine les différents aspects du contrôle d'accès ainsi qu'un fichier `sample.conf` avec des idées sur la modification des fichiers de configuration spécifiques à des services donnés qui se trouvent dans le répertoire `/etc/xinetd.d/`.
- `man 5 hosts_access` — La page de manuel relative aux fichiers de contrôle d'accès des hôtes des enveloppeurs TCP.
- `man hosts_options` — La page de manuel relative aux champs d'options des enveloppeurs TCP.
- `man xinetd.conf` — La page de manuel énumérant les options de configuration de `xinetd`.
- `man xinetd` — La page de manuel relative au démon du super-service `xinetd`.

15.5.2. Sites Web utiles

- http://www.xinetd.org — La page d'accueil de `xinetd`, contenant avec des exemples de fichiers de configuration, une liste complète des fonctions et un FAQ très riche.
- http://www.macsecurity.org/resources/xinetd/tutorial.shtml — Un tutoriel complet décrivant les nombreuses manières différentes de modifier les fichiers de configuration par défaut de `xinetd` afin qu'ils correspondent à des but de sécurité spécifiques.

15.5.3. Livres sur le sujet

- *Guide de sécurité de Red Hat Linux* ; Red Hat, Inc. — Fournit un aperçu de la sécurité en matière de station de travail, serveur et réseau et contient des suggestions spécifiques quant aux enveloppeurs TCP et au service `xinetd`.
- *Hacking Linux Exposed* de Brian Hatch, James Lee et George Kurtz; Osbourne/McGraw-Hill — Une excellente ressource sur la sécurité contenant des informations sur les enveloppeurs TCP et le service `xinetd`.



Chapitre 16. iptables

Avec Red Hat Linux sont installés des outils permettant le *filtrage de paquets* réseau — le processus contrôlant les paquets réseau lorsqu'ils entrent, traversent et sortent de la pile réseau dans le noyau. Les noyaux antérieurs à la version 2.4 utilisaient `ipchains` pour le filtrage de paquets et faisaient appel à des listes de règles appliquées aux paquets à chaque étape du processus de filtrage. Avec le noyau version 2.4 a été mis service `iptables` (aussi appelé *netfilter*), qui est semblable à la commande `ipchains` si ce n'est qu'elle multiplie les potentialités et le degré de contrôle disponible lors du filtrage de paquets.

Ce chapitre décrit en détail les principes des techniques de filtrage de paquets en expliquant les différences entre `ipchains` et `iptables`, en présentant les différentes options disponibles avec `iptables` et en montrant comment maintenir l'intégrité des règles de filtrage entre chaque démarrage de votre système.

Pour obtenir des instructions sur la construction de règles `iptables` ou sur la configuration d'un pare-feu basé sur ces règles, reportez-vous à la Section 16.5.

**Avertissement**

Le mécanisme de pare-feu par défaut dans le noyau 2.4 est `iptables`, mais `iptables` ne peut pas être utilisé si `ipchains` est déjà à mis en oeuvre. Si `ipchains` sont présentes au démarrage, le noyau annoncera une erreur et ne pourra pas lancer `iptables`.

Ces messages d'erreur lors du démarrage n'affectent pas la fonctionnalité d'`ipchains`.

16.1. Filtrage de paquets

Les informations se déplacent à l'intérieur d'un réseau sous la forme de *paquets*. Un paquets réseau représente un ensemble de données de dimension et format particuliers. Afin de transmettre un fichier sur un réseau, l'ordinateur faisant l'envoi doit d'abord diviser le fichier en paquets en fonction des règles du protocole réseau. Chacun de ces paquets contient une petite partie des données du fichier. À la réception de la transmission l'ordinateur cible reconstruit le fichier à l'aide des paquets.

Chaque paquet dispose de ses propres informations de navigation lui permettant de se déplacer sur le réseau jusqu'à sa destination finale. Le paquet est capable d'indiquer entre autres sa provenance, sa destination et son identité aux ordinateurs rencontrés tout au long de son parcours, ainsi qu'à l'ordinateur vers lequel il se dirige. La plupart des paquets servent au transport de données, bien que certains protocoles les utilisent pour des tâches particulières. Par exemple, le protocole de transmission connu sous l'abréviation *TCP ('Transmission Control Protocol')*, utilise un paquet nommé SYN qui ne contient aucune donnée et sert à établir une communication entre deux systèmes.

Parmi ses nombreux rôles, le noyau Linux à la possibilité de filtrer des paquets en choisissant de laisser pénétrer certains d'entre eux dans le système et de bloquer les autres. La version 2.4 du noyau contient les trois *tables* ou *listes de règles* suivantes:

- `filter` — table par défaut pour le traitement des paquets réseau.
- `nat` — table utilisée pour modifier les paquets qui créent une nouvelle connexion.
- `mangle` — table utilisée pour la modification de types spécifiques de paquets.

Chacune de ces tables comporte à son tour un groupe de *chaînes* qui correspondent aux actions effectuées par netfilter sur le paquet.

Les chaînes de la table `filter` sont les suivantes:

- *INPUT* — cette chaîne s'applique aux paquets reçus via une interface réseau.
- *OUTPUT* — cette chaîne s'applique aux paquets expédiés par la même interface réseau qui a reçu les paquets.
- *FORWARD* — cette chaîne s'applique aux paquets reçus par une interface réseau et expédiés par une autre interface.

Les chaînes pour la table `nat` sont les suivantes:

- *PREROUTING* — cette chaîne modifie les paquets reçus via une interface réseau lorsqu'ils arrivent.
- *OUTPUT* — cette chaîne modifie les paquets générés localement avant qu'ils ne soient routés par une interface réseau.
- *POSTROUTING* — cette chaîne modifie les paquets avant qu'ils ne soient expédiés par une interface de réseau.

Les chaînes pour la table `mangle` sont les suivantes:

- *PREROUTING* — cette chaîne modifie les paquets reçus par une interface de réseau avant qu'ils ne soient routés.
- *OUTPUT* — cette chaîne modifie les paquets générés localement avant qu'ils ne soient routés via une interface réseau.

Chaque paquet de réseau reçu ou expédié par un système Linux est soumis à au moins une table.

Chaque paquet, avant de sortir d'une chaîne, peut être soumis à un très grand nombre de règles. La structure et le rôle de ces règles peuvent changer, mais elles visent généralement à identifier un paquet en provenance ou à destination d'une adresse IP donnée ou d'un groupe d'adresses, lors de l'utilisation d'un protocole ou d'un service de réseau particulier.

Indépendamment de leur destination, lorsque les paquets correspondent à une règle précise d'une des tables, ils se voient assigner une *cible* font l'objet d'une certaine action. Si la règle spécifie une cible de type `ACCEPT`, un paquet est accepté après avoir été contrôlé avec succès, et peut procéder vers sa destination en évitant le reste des contrôles de la règle. Si une règle spécifie une cible de type `DROP` le paquet est abandonné et se voit refuser l'accès au système; rien n'est donc renvoyé à l'hôte qui a expédié le paquet. Si une règle spécifie une cible de type `QUEUE`, le paquet est mis en attente dans l'espace-utilisateur (ou 'user-space'). Finalement, si une règle spécifie une cible de type `REJECT`, le paquet est "abandonné" mais par rejet et dans ce cas, un "paquet d'erreur" est renvoyé à l'expéditeur.

Chaque chaîne dispose d'une politique par défaut pour accepter (`ACCEPT`), abandonner (`DROP`), rejeter (`REJECT`) ou mettre en attente (`QUEUE`). Si aucune des règles présentes dans la chaîne ne s'applique au paquetage, celui-ci est traité en fonction de la politique par défaut de la chaîne.

La commande `iptables` permet de configurer ces tables et d'en créer de nouvelles si nécessaire.

16.2. Les différences entre `iptables` et `ipchains`

Au premier abord, `ipchains` et `iptables` semblent assez similaires. Les deux méthodes de filtrage de paquets font appel à des chaînes de règles actives à l'intérieur du noyau Linux pour décider non seulement du type de paquets autorisés à entrer ou sortir du système, mais également du traitement des paquets qui répondent à certaines règles. Cependant, la commande `iptables` représente une manière plus flexible de filtrer les paquets en donnant à l'administrateur système un degré de contrôle plus élevé sans pour autant ajouter un degré plus élevé de complexité.

Ainsi, les utilisateurs à l'aise avec la commande `ipchains` devront tenir compte des différences importantes existant entre les commandes `ipchains` et `iptables` avant d'essayer de se servir de `iptables`:

- *Sous `iptables`, chaque paquet filtré est traité en utilisant les règles d'une seule chaîne, plutôt que celles de chaînes multiples.* Par exemple, un paquet identifié comme FORWARD pénétrant dans un système à l'aide de `ipchains` devrait passer à travers les chaînes INPUT, FORWARD et OUTPUT afin de pouvoir poursuivre sa progression vers sa destination. Toutefois, `iptables` envoie les paquets uniquement à la chaîne INPUT s'ils sont destinés au système local et vers la chaîne OUTPUT, s'ils ont été créés par le système local. Pour cette raison, il très important de bien placer la règle destinée au contrôle d'un paquet spécifique dans la bonne règle qui détectera vraiment le paquet.
- *La cible DENY a été remplacée par la cible DROP.* Dans `ipchains`, les paquets qui satisfaisaient les critères d'une règle dans une chaîne pouvaient être dirigés vers la cible DENY. Cette cible doit être substituée par une cible DROP `iptables`.
- *Lorsque des options sont placées dans une règle, l'ordre de placement est primordial.* Auparavant, avec `ipchains`, cet ordre d'écriture importait peu. La commande `iptables` elle, utilise une syntaxe plus stricte. Par exemple, dans les commandes `iptables` le type de protocole (ICMP, TCP ou UDP) doit être précisé avant de spécifier les ports d'origine ou de destination. ports.
- *Lorsque le type d'interface réseau à utiliser dans une règle doit être précisé, seules des interfaces d'entrée (option `-i`) peuvent être employées avec les chaînes INPUT ou FORWARD et des interfaces de sortie (option `-o`) avec les chaînes FORWARD ou OUTPUT.* Ceci est nécessaire d'une part parce que les chaînes OUTPUT ne sont plus utilisées par les interfaces d'entrée et d'autre part, parce que les chaînes INPUT ne sont pas vues par les paquets se déplaçant au travers des interfaces de sortie.

Les précisions ci-dessus ne constituent en aucun cas une liste compréhensive des changements apportés; en effet, `iptables` représente fondamentalement un filtre réseau réécrit. Pour obtenir des informations plus spécifiques, reportez-vous au document *Linux 2.4 Packet Filtering HOWTO* qui se trouve dans la Section 16.5.

16.3. Options utilisées avec les commandes `iptables`

Les règles permettant le filtrage de paquets par le noyau sont mise en oeuvre en exécutant la commande `iptables`. Lorsque vous utilisez la commande `iptables`, vous devez spécifier les options suivantes:

- *Type de paquet* — stipule le type de paquets que la commande filtre.
- *Origine/Destination du paquet* — spécifie les paquets que la commande filtre sur la base de l'origine ou de la destination du paquets.
- *Cible* — stipule l'action à appliquer sur les paquets remplissant les critères évoqués ci-dessus.

Les options utilisées avec une règle `iptables` donnée doivent être logiquement groupées, sur la base du but et des conditions de la règle générale, afin que la règle soit valide.

16.3.1. Tables

Un des points forts de `iptables` réside dans la possibilité d'utiliser des tables multiples pour décider du sort d'un paquet donné. Grâce à la nature flexible de `iptables`, des tables spécifiques peuvent être créées et enregistrées dans le répertoire `/lib/modules/`<*version-du-noyau*>`/kernel/net/ipv4/netfilter/` où <*version-du-noyau*> correspond au numéro de version du noyau.

La table par défaut, appelée `filter`, contient les chaînes standard intégrées INPUT, OUTPUT et FORWARD. Ceci est assez semblable aux chaînes standard en usage avec `ipchains`. Toutefois, `iptables` possède aussi par défaut deux tables supplémentaires qui effectuent des opérations de filtrage de paquets spécifiques. La table `nat` peut être utilisée pour modifier les adresses d'origine et de destination enregistrées dans les paquets alors que la table table `mangle` permet de modifier des paquets selon des méthodes particulières.

Chaque table contient certes des chaînes par défaut dont le but est d'exécuter des tâches selon l'objectif même de la table, mais il est également possible de définir de nouvelles chaînes dans chaque table.

16.3.2. Structure

Beaucoup de commandes `iptables` ont la structure suivante:

```
iptables [-t <nom-de-table>]
<commande>
<nom-de-chaîne>
<paramètre-1> \
        <option-1>
<paramètre-n>
<option-n>
```

Dans cet exemple, l'option <*nom-table*> permet à l'utilisateur de sélectionner une autre table que la table par défaut `filter` à utiliser avec cette commande. L'option <*commande*> stipule une action spécifique à accomplir, telle que l'ajout ou l'élimination d'une règle spécifiée par <*nom-de-chaîne*>. après l'option <*nom-de-chaîne*> se trouve une paire de paramètres et d'options servant à définir l'action à entreprendre lorsqu'un paquet correspond au critères de la règle.

En examinant la structure d'une commande `iptables`, il est important de se rappeler que contrairement aux autres commandes, la longueur et la complexité d'une commande `iptables` varie en fonction de son objectif. Une simple commande servant à éliminer une règle d'une chaîne peut être très courte, alors qu'une commande servant à filtrer les paquets d'un sous-réseau faisant appel à un certain nombre de paramètres et d'options sera plutôt longue. Lors de la création de commandes `iptables`, il est important de se rappeler que certains paramètres et options peuvent nécessiter la création de paramètres et options supplémentaires pour mieux définir la requête de l'option précédente. Pour écrire une règle valide, cette chaîne d'actions doit continuer jusqu'à ce que chaque paramètre et option nécessitant une autre série d'options soit satisfait.

Entrez la commande `iptables -h` pour obtenir une liste exhaustive de structures de commandes `iptables`.

16.3.3. Commandes

Les commandes donnent à `iptables` l'instruction d'exécuter une action spécifique. Seule une commande est autorisée par chaîne de commande `iptables`. À l'exception de la commande d'aide, toutes les autres commandes doivent être écrites en majuscules.

Les commandes `iptables` disponibles sont les suivantes:

- `-A` — ajoute une règle `iptables` à la fin d'une chaîne donnée. On l'utilise pour ajouter simplement une règle lorsque l'ordre des règles à l'intérieur de la chaîne n'est pas primordial.
- `-C` — contrôle une règle donnée avant de l'ajouter à la chaîne spécifiée par l'utilisateur. Cette commande peut vous aider à écrire des règles `iptables` compliquées en vous indiquant les paramètres et options supplémentaires à établir.

- -D — élimine une règle à l'intérieur d'une chaîne donnée de façon numérique (comme par exemple en utilisant 5,pour la cinquième règle d'une chaîne). Il est également possible de taper la règle complète et iptables effacera la règle dans la chaîne correspondante.
- -E — sert à changer le nom d'une chaîne spécifiée par un utilisateur. Cette option n'affecte en aucun cas la structure de la table.
- -F — supprime la chaîne sélectionnée, entraînant par là-même, l'élimination de toutes les règles de la chaîne. Si aucune chaîne n'est spécifiée, cette commande supprime chaque règle de chaque chaîne.
- -h — fournit une liste des structures de commande, ainsi qu'un bref résumé des paramètres et options des commandes.
- -I — insère une règle à l'intérieur d'une chaîne, à un point précis, spécifié par une valeur paire définie par l'utilisateur. Si aucun numéro n'est spécifié, iptables placera la commande au tout début de la chaîne.


Attention

Prêtez particulièrement attention à l'option (-A or -I) utilisée lors de l'ajout d'une règle. L'ordre dans lequel les règles apparaissent dans une chaîne est très important quand il s'agit de définir quelles règles appliquer à quels paquets.

- -L — établit la liste complète des règles dans la chaîne indiquée après la commande. Pour obtenir une liste de toutes les règles de toutes les chaînes contenues dans la table par défaut, filter, ne précisez ni chaîne, ni table. Sinon, la syntaxe à utiliser pour établir la liste des règles contenues dans une chaîne donnée, d'une table précise, doit être la suivante:

  ```
  iptables -L <nom-de-chaîne> -t
  <nom-de-table>
  ```

  Des options puissantes pour la commande -L fournissant le nombre de règles et permettant une description très détaillée de ces dernières sont décrites dans la Section 16.3.7.
- -N — crée une nouvelle chaîne avec un nom spécifié par l'utilisateur.
- -P — définit la politique par défaut d'une chaîne donnée, de sorte que lorsque des paquets traversent une chaîne entière sans satisfaire à une règle, ils seront envoyés à une cible donnée, telle que ACCEPT ou DROP.
- -R — remplace une règle dans une chaîne donnée. Il est impératif d'utiliser un numéro de règle après le nom de chaîne. La première règle dans une chaîne correspond à la règle numéro un.
- -X — supprime une chaîne spécifiée par un utilisateur. L'élimination d'une chaîne intégrée de toute table n'est pas permise.
- -Z — remet à zéro les compteurs d'octets et de paquets pour toutes les chaînes pour une table spécifique.

16.3.4. Paramètres

Une fois que certaines commandes iptables ont été spécifiées (y compris celles utilisées pour l'ajout, l'élimination, l'insertion ou le remplacement de règles à l'intérieur d'une chaîne donnée), il est nécessaire d'ajouter d'autres paramètres pour la construction d'une règle de filtrage de paquets.

- -c effectue une remise à zéro des compteurs pour une règle donnée. Ce paramètre accepte les options PKTS (paquets) et BYTES (octets) pour indiquer le compteur à remettre à zéro.
- -d — définit le nom d'hôte du destinataire, l'adresse IP ou le réseau du paquetage qui correspondra à la règle. Lors de la vérification de concordance réseau, les formats adresses IP/masque réseau suivants sont pris en charge:

  - *N.N.N.N/M.M.M.M* — où *N.N.N.N* correspond à la plage de l'adresse IP et *M.M.M.M* au masque réseau.
  - *N.N.N.N/M* — où *N.N.N.N* correspond à la plage de l'adresse IP et *M* au masque réseau.

- `-f` applique cette règle uniquement aux paquets fragmentés.

  En insérant l'option `!` après ce paramètre, seuls les paquets non-fragmentés seront contrôlés.

- `-i` — règle l'interface réseau d'entrée, telle que `eth0` ou `ppp0`. Avec `iptables`, ce paramètre optionnel ne peut être utilisé qu'avec des chaînes INPUT et FORWARD, lorsqu'elles sont utilisées avec la table `filter` et la chaîne PREROUTING avec les tables `nat` et `mangle`.

  Ce paramètre prend également en charge les options spéciales suivantes:

  - `!` — donne l'instruction à ce paramètre de ne pas comparer, signifiant que n'importe quelle interface spécifiée est exclue de cette règle.
  - `+` — un caractère générique (ou 'wildcard') utilisé pour comparer toutes les interfaces qui correspondent à une chaîne particulière. Par exemple, le paramètre `-i eth+` appliquerait cette règle à n'importe quelle interface Ethernet, mais ne prendrait pas en compte les autres interfaces, comme `ppp0`.

  Si le paramètre `-i` est utilisé sans qu'aucune interface ne soit spécifiée, alors toutes les interfaces sont affectées par la règle.

- `-j` — donne à `iptables` l'instruction de passer directement à une cible donnée lorsqu'un paquetage correspond à une règle particulière. Les cibles autorisées après l'option `-j` incluent les options standard `ACCEPT`, `DROP`, `QUEUE`, et `RETURN`, ainsi que des options étendues qui sont disponibles dans des modules chargés par défaut avec le paquetage RPM de commandes Red Hat Linux nommé `iptables`, comme, entre autres, `LOG`, `MARK` et `REJECT`. Consultez la page de manuel relatives à `iptables` pour obtenir plus d'informations sur les cibles.

  il est également possible de diriger un paquet correspondant à une règle vers une chaîne définie par l'utilisateur, située en dehors de la chaîne courante, afin que d'autres règles puissent être appliquées à ce paquet.

  Si aucune cible n'est spécifiée, le paquet continue sans qu'aucune autre action ne soit entreprise. Ceci étant, le compteur de cette règle avance tout de même d'un point car le paquet correspond à la règle spécifiée.

- `-o` — règle l'interface de sortie pour une règle donnée et ne peut être utilisée qu'avec des chaînes OUTPUT et FORWARD dans la table `filter` et la chaîne POSTROUTING dans les tables `nat` et `mangle`. Les options de ce paramètre sont les mêmes que pour les paramètres relatifs aux interfaces réseau d'entrée (`-i`).

- `-p` — règle le protocole IP pour la règle, qui peut être `icmp`, `tcp`, `udp` ou `all`, pour correspondre à tous les protocoles possibles. De plus, tout protocole inclus dans `/etc/protocols` peuvent également être employés. Si l'option est omise lors de la création de la règle, l'option `all` est considérée comme étant la valeur par défaut.

- `-s` — définit l'origine d'un paquet particulier en utilisant la même syntaxe que pour le paramètre de destination (`-d`).

16.3.5. Options de concordance

Différents protocoles réseau offrent des options de contrôle de concordance spécifiques qui peuvent être configurées de manière à comparer un paquet donné utilisant ce protocole. Évidemment, il est nécessaire d'identifier préalablement le protocole en question dans la commande `iptables` à l'aide de l'option `-p tcp` <*nom-du-protocole*> (où <*nom-du-protocole*> correspond au protocole cible), afin que ces options soient disponibles.

16.3.5.1. Protocole TCP

Les options de concordance disponibles pour le protocole TCP (`-p tcp`) sont les suivantes:

- `--dport` — indique le port de destination pour le paquet. Vous pouvez utiliser un nom de service de réseau (comme `www` ou `smtp`), un numéro de port ou une plage de numéros de port pour configurer cette option. Pour parcourir les noms et alias de services réseau et les numéros de port utilisés, affichez le fichier `/etc/services` L'option de concordance `--destination-port` est identique à l'option `--dport`.

  Pour indiquer une plage précise de numéros de port, il suffit de séparer les numéros par le symbole des deux points (`:`), comme dans l'exemple suivant: `-p tcp --dport 3000:3200`. La plus grande plage possible est `0:65535`.

  Utilisez un point d'exclamation (`!`) après l'option `--dport` pour donner `iptables` l'instruction comparer tous les paquets qui *n'utilisent pas* ce service de réseau ou port.
- `--sport` — indique le port d'origine du paquet, en utilisant les mêmes options que `--dport`. L'option de concordance `--source-port` est identique à l'option `--sport`.
- `--syn` s'applique à tous les paquets TCP, appelés communément *paquets SYN*, conçus pour initier la communication. Aucun paquet transportant des données de charge utile n'est touché. En plaçant un point d'exclamation (`!`) comme indicateur après l'option `--syn`, tous les paquets non-SYN seront comparés.
- `--tcp-flags` — permet la comparaison avec une règle de paquets TCP ayant une taille en octets ou des indicateurs spécifiques. L'option de concordance `--tcp-flags` accepte deux paramètres. Le premier paramètre est le masque, qui définit l'indicateur à examiner pour le paquet. Le second se rapporte aux indicateurs qui doivent être définis afin que la concordance puisse avoir lieu.

  Les indicateurs disponibles sont les suivants:

  - `ACK`
  - `FIN`
  - `PSH`
  - `RST`
  - `SYN`
  - `URG`
  - `ALL`
  - `NONE`

  Par exemple, une règle `iptables` contenant `-p tcp --tcp-flags ACK,FIN,SYN SYN` ne comparera que les paquets TCP ayant l'indicateur SYN défini et les indicateurs ACK et FIN non-définis.

  L'utilisation d'un point d'exclamation (`!`) après `--tcp-flags` inverse l'effet de l'option de concordance.
- `--tcp-option` essaie de comparer des options spécifiques à TCP qui peuvent être définies dans un paquet donné. Cette option de concordance peut aussi être inversée en utilisant un point d'exclamation (`!`).

16.3.5.2. Protocole UDP

Les options de concordance suivantes s'appliquent au protocole UDP (`-p udp`) :

- `--dport` — indique le port de destination du paquet UDP, en utilisant le nom du service, le numéro de port ou une plage de numéros de port. L'option de concordance `--destination-port` est identique à l'option `--dport`. Reportez-vous à l'option de concordance `--dport` dans la Section 16.3.5.1 pour obtenir des informations sur les modalités d'utilisation de cette option. option.
- `--sport` — indique le port d'origine du paquet UDP en utilisant le nom de service, le numéro de port ou une plage de numéros de port. L'option de concordance `--source-port` est identique à l'option `--sport`. Reportez-vous à l'option de concordance `--sport` dans la Section 16.3.5.1 pour obtenir des informations sur les modalités d'utilisation de cette option.

16.3.5.3. Protocole ICMP

Les options de concordance suivantes sont disponibles pour le protocole Internet Control Message Protocol (ICMP) (`-p icmp`):

- `--icmp-type` définit le nom ou le numéro du type d'ICMP à comparer avec cette règle. Une liste de noms ICMP valides est disponible en tapant la commande `iptables -p icmp -h`.

16.3.5.4. Modules avec options de concordance supplémentaires

Des options de concordance supplémentaires sont également disponibles par l'entremise des modules chargés par la commande `iptables`. Pour utiliser un module d'option de concordance, chargez le module en l'appelant par son nom à l'aide de l'option `-m`, comme par exemple: `-m` <*nom-du-module*> (où <*nom-du-module*> correspond au nom du module).

Un nombre important de modules est disponible par défaut. Il est même possible de créer vos propres modules pour fournir des options de concordance supplémentaires pour une fonctionnalité accrue.

Il existe de nombreux modules, mais seuls les plus fréquents sont abordés ici.

- `limit` module — permet de limiter le nombre de paquets qui sont comparés à une règle donnée. Ceci se révèle tout particulièrement pratique lors de la concordance avec des règles de journalisation, afin d'éviter que les résultats n'entraînent l'invasion de vos journaux par des messages répétitifs ou ne consomment trop de ressources système.

  Le module `limit` permet les options suivantes:

  - `--limit` — limite le nombre de concordances dans un espace-temps donné, grâce à un modificateur de nombre et de temps paramétré sous la forme suivante: <*nombre*>/<*temps*>. Par exemple, en écrivant `--limit 5/hour`, une règle effectue son contrôle de concordance seulement cinq fois par heure.

    Si aucun modificateur de nombre ou temps n'est précisé, une valeur par défaut de `3/hour` sera retenue.
  - `--limit-burst` — limite le nombre de paquets pouvant être comparés à une règle, à un moment donné. Cette option est à utiliser conjointement avec l'option `--limit` et accepte un numéro pour en définir le seuil.

    Si aucun numéro n'est indiqué, seulement cinq paquets sont en mesure d'être contrôlés à la règle.

- module `state` — permet la concordance d'état.

  Ce module `state` permet les options suivantes:

  - `--state` — compare un paquet avec les états de connexion suivants:
    - `ESTABLISHED` — le paquet contrôlé est associé à d'autres paquets dans une connexion établie.

- `INVALID` — le paquet contrôlé ne peut être associé à une connexion connue.
- `NEW` — le paquet contrôlé crée une nouvelle connexion ou fait partie d'une connexion à double sens qui n'a pas encore été vue.
- `RELATED` — le paquet contrôlé commence une nouvelle connexion liée d'une façon ou d'une autre à une connexion existante.

Ces états de connexion peuvent être employés de concert avec d'autres à condition qu'ils soient séparés par des virgules, comme par exemple: `-m state --state INVALID,NEW`.

- `mac` module — permet la concordance d'une adresse MAC matérielle.

  Le module `mac` permet l'option suivante:

  - `--mac-source` — compare une adresse MAC de la carte d'interface réseau qui a envoyé le paquet. Pour exclure une adresse MAC d'une règle, placez un point d'exclamation (`!`) après l'option de concordance `--mac-source`.

Pour obtenir des informations sur d'autres options de concordance disponibles avec les modules, reportez-vous à la page de manuel de `iptables`.

16.3.6. Options de cible

Une fois que la concordance d'un paquet a été contrôlée par une règle spécifique, cette dernière peut diriger le paquet vers un certain nombre de cibles qui décideront de son traitement et, si possible, entreprendront des actions supplémentaires. Chaque chaîne possède une cible par défaut qui est utilisée si aucune des règles de la chaîne ne correspond à un paquet ou si aucune des règles qui correspondent à un paquet ne spécifie de cible particulière.

Ci-dessous figurent les cibles standards:

- <*chaîne-spécifiée-par-l'utilisateur*> — remplacez <*chaîne-spécifiée-par-l'utilisateur*> par le nom d'une chaîne définie par l'utilisateur au sein de cette table. Cette cible transmet le paquet à la chaîne cible.
- `ACCEPT` — permet au paquet de continuer sa progression vers sa destination (ou une autre chaîne si sa configuration l'y oblige).
- `DROP` — abandonne le paquet répondre au demandeur. Le système ayant expédié ce paquet n'est pas informé de l'échec de l'opération.
- `QUEUE` — le paquet est mis en attente et sera traité par une application de l'espace-utilisateur (user-space).
- `RETURN` — arrête le contrôle du paquet en fonction des règles en vigueur dans la chaîne actuelle. Si le paquet avec la cible `RETURN` correspond à une certaine règle appelée depuis une autre chaîne, le paquet est renvoyé à la première chaîne pour la continuation de son contrôle au point où il s'était arrêté. Dans le cas où la règle `RETURN` est utilisée dans une chaîne intégrée et que le paquet ne peut pas revenir vers la chaîne précédente, la cible appliquée par défaut décide alors de l'action à entreprendre.

Outre ces cibles standards, plusieurs autres cibles peuvent être utilisées avec des extensions appelées *modules cibles*, qui fonctionnent d'une manière semblable aux modules d'options de concordance (reportez-vous à la Section 16.3.5.4.

Il existe de nombreux modules cibles étendus; la plupart d'entre eux s'appliquent à des tables ou à des situations spécifiques. Ci-dessous figurent certains des modules cibles les plus répandus, inclus par défaut dans Red Hat Linux :

- LOG — journalise tous les paquets correspondant à cette règle. Étant donné que les paquets sont journalisés par le noyau, le fichier /etc/syslog.conf détermine l'emplacement où ces entrées sont enregistrées. Par défaut, elles sont placées dans le fichier /var/log/messages.

  Différentes options peuvent être utilisées après la cible LOG pour spécifier le processus de journalisation:

  - --log-level — détermine le niveau de priorité d'un événement de journalisation. Une liste de niveaux de priorité est disponible dans la page de manuel de syslog.conf.
  - --log-ip-options spécifie que toute option indiquée dans l'en-tête d'un paquet IP est journalisée.
  - --log-prefix — ajoute une chaîne comportant au maximum 29 caractère avant la ligne du journal, lorsqu'elle est écrite. Cette option est utile lors de l'écriture de filtres syslog à utiliser conjointement avec la journalisation de paquets.
  - --log-tcp-options — indique que toute option précisée dans l'en-tête d'un paquet TCP est journalisée.
  - --log-tcp-sequence écrit le numéro de séquence TCP relatif au paquet dans le journal.

- REJECT — renvoie un paquet d'erreur au système ayant expédié le paquet et abandonne le paquet.

  La cible REJECT accepte une option --reject-with <*type*> (où <*type*> correspond au type de rejet) qui permet de d'inclure des informations plus détaillées avec le paquet d'erreur. Le message d'erreur port-unreachable (impossible d'atteindre le port) représente le <*type*> d'erreur par défaut envoyée si aucune autre option n'est utilisée. Pour obtenir une liste complète des options <*type*> disponibles, consultez la page de manuel relative à iptables.

D'autres extensions de cibles, dont bon nombre étant très utiles pour le masquage d'IP (ou masquerading) faisant appel à la table nat ou avec la modification de paquets à l'aide de la table mangle, se trouvent dans la page de manuel iptables.

16.3.7. Options de listage

La commande de listage par défaut, iptables -L, forunit un aperçu très élémentaire des chaînes actuelles contenues dans la table de filtres par défaut. Des options supplémentaires donnent plus d'informations:

- -v affiche une sortie prolixe, indiquant le nombre de paquets et octets lus par chaque chaîne, le nombre de paquets et d'octets contrôlés par chaque règle et l'identité des interfaces liées aux règles.
- -x présente les nombres selon leur valeur exacte. Dans un système très chargé, le nombre de paquets et d'octets vus par une chaîne donnée peut être abrégé en utilisant K (milliers), M (millions) et G (milliards) à la fin du nombre. Cette option oblige l'affichage du nombre réel.
- -n affiche les adresses IP et les numéros de port de façon numérique, plutôt que d'utiliser le nom d'hôte et le format du service de réseau.
- --line-numbers énumère les règles dans chaque chaîne à côté de leur ordre numérique dans la chaîne. Cette option est utile lorsque l'on tente d'éliminer une règle donnée dans une chaîne ou de localiser l'emplacement d'une règle à insérer dans une chaîne.
- -t — spécifie un nom de table.

16.4. Stockage de l'information iptables

Les règles créées avec la commande iptables sont stockées en mémoire. Si le système est redémarré après la configuration des différentes règles iptables, elles seront perdues. Pour que des règles de filtrage réseau soient conservées lors d'un redémarrage, elles doivent être enregistrées. Pour ce faire, connectez-vous en tant que super-utilisateur (ou root) et tapez:

```
/sbin/service iptables save
```

Cette commande exécute le script initial (init script) iptables, qui lancer le programme /sbin/iptables-save et enregistre la configuration actuelle de iptables dans le fichier /etc/sysconfig/iptables. Ce fichier ne devrait être lisible que par le super-utilisateur.

Au prochain démarrage, le script initial iptables fera appliquer les règles enregistrées dans /etc/sysconfig/iptables grâce à la commande /sbin/iptables-restore.

Alors qu'il est toutjours préférable de tester une nouvelle règle iptables avant de l'enregistrer dans le fichier /etc/sysconfig/iptables, il est possible de copier des règles iptables dans ce fichier à partir d'une version de ce fichier provenant d'un autre ordinateur. Cette opération permet de distribuer facilement un ensemble de règles iptables à de multiples ordinateurs.



Important

Si vous distribuez le fichier /etc/sysconfig/iptables vers d'autres machines, il suffit de taper /sbin/service iptables restart pour que ces nouvelles règles soient mises en oeuvre.

16.5. Sources d'informations supplémentaires

Veuillez consulter les informations ci-dessous pour des informations supplémentaires sur le filtrage de paquets avec iptables.

16.5.1. Documentation installée

- man iptables — contient une description complète des différents paramètres et commandes ainsi que d'autre options.

16.5.2. Sites Web utiles

- http://netfilter.samba.org — contient une série d'informations sur iptables, y compris un FAQ traitant de problèmes spécifiques et un certain nombres de guides rédigés par Rusty Russell, le responsable du pare-feu IP de Linux. Les documents HOWTO couvrent des sujets de base, tels que les concepts élémentaires de mise en réseaux, les techniques de filtrage de paquets avec le noyau 2.4 et les configurations NAT.
- http://www.linuxnewbie.org/nhf/Security/IPtables_Basics.html — une présentation simple concernant le déplacement de paquets dans le noyau Linux, ainsi qu'une introduction à la construction de commandes iptables simples.
- http://www.redhat.com/support/resources/networking/firewall.html — cette page contient plusieurs liens mis à jour vers diverses ressources traitant du filtrage de paquets.



Chapitre 17.
Kerberos

Kerberos est un protocole d'authentification réseau créé par MIT et utilisant une cryptographie à clés secrètes pour authentifier les utilisateurs de services d'un réseau — éliminant par là-même, le besoin de transmettre des mots de passe sur le réseau. L'utilisation de Kerberos pour authentifier des utilisateurs avant qu'ils ne puissent utiliser les services du réseau, permet d'éviter que des utilisateurs non-autorisés essaient d'intercepter des mots de passe sur le réseau en surveillant le trafic.

17.1. Les avantages de Kerberos

La plupart des systèmes de réseau conventionnels utilisent des procédures d'authentification par mot de passe. Dans le cadre des telles procédures, un utilisateur doit s'authentifier auprès d'un certain serveur réseau précis en fournissant son nom d'utilisateur et son mot de passe. Regrettablement, la transmission des informations d'authentification pour de nombreux services s'effectue de façon non-cryptée. Pour qu'une telle procédure soit sécurisée, le réseau doit être inaccessible aux utilisateurs externes, et il est essentiel de pouvoir faire confiance à tous les ordinateurs et utilisateurs sur le réseau.

Même si tel est le cas, une fois qu'un réseau est connecté à l'Internet, on ne peut plus supposer que le réseau demeure sécurisé. Il suffit à un pirate qui obtient l'accès au réseau d'utiliser un simple analyseur de paquets, aussi connu sous le nom de renifleur de paquets, pour intercepter des noms d'utilisateur et des mots de passes envoyés en texte clair. De ce fait, les comptes utilisateurs et l'intégrité de toute l'infrastructure de sécurité sont remis en cause.

Le but essentiel de Korberos est d'éviter la transmission de mots de passe non-cryptés à travers le réseau. Lorsque Korberos est utilisé correctement, il élimine de façon efficace la menace que posent sur un système les renifleurs de paquets.

17.1.1. Désavantages de Kerberos

Kerberos permet certes d'éliminer une menace commune pour la sécurité, mais son implémentation peut être difficile pour de multiples raisons:

- La migration de mots de passe utilisateur d'une base de données de mots de passe UNIX standard, comme `/etc/passwd` ou `/etc/shadow`, vers une base de données de mots de passe Kerberos peut être relativement longue car il n'existe aucun mécanisme automatique permettant d'effectuer cette tâche. Pour de plus amples informations sur le sujet, consultez le point numéro 2.23 dans le FAQ de Kerberos qui se trouve à l'URL suivante:

  http://www.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html.
- Kerberos n'est que partiellement compatible avec le système PAM ('Pluggable Authentication Module', module d'authentification enfichable) utilisé par la plupart des serveurs exécutant Red Hat Linux. Pour plus d'informations, reportez-vous à la Section 17.4.
- Pour qu'une application utilise Kerberos, ses sources doivent être modifiées afin de faire les appels appropriés dans les bibliothèques Kerberos. Pour certaines applications, ceci peut poser de nombreux problèmes en raison de la taille et de la conception de l'application. Pour d'autres applications qui ne sont pas compatibles, des modifications doivent être apportées à la manière dont les serveurs et les clients communiquent entre eux. Là encore, il se peut que des modifications importantes au niveau de la programmation soient nécessaires. Les applications dont les sources ne sont pas accessibles et dont le support pour Kerberos n'est pas disponible sont celles posant généralement le plus de problèmes.

- Kerberos suppose que vous êtes des utilisateurs sécurisés utilisant un hôte non-sécurisé sur un réseau non-sécurisé. Son but primaire est d'empêcher que des mots de passe en texte clair ne soient envoyés à travers ce réseau. Toutefois, si quelqu'un d'autre que l'utilisateur normal a physiquement accès à l'hôte qui émet les tickets utilisés pour l'authentification — nommé *centre de distribution de clés* ('key distribution center' *KDC*) — tout le système d'authentification Kerberos est menacé d'être compromis.
- Avec un solution Kerberos, c'est tout ou rien. Si vous décidez d'utiliser Kerberos sur votre réseau, rappelez-vous bien que tout mot de passe transmis à un service qui n'utilise pas Kerberos pour l'authentification risque d'être intercepté par des renifleurs de paquets. Dans de telles conditions, votre système ne tirera aucun avantage de l'utilisation de Korberos. Afin de sécuriser votre réseau avec Korberos, vous devez soit utiliser des versions *'kerberisées'* de *toutes* les applications client/serveur qui envoient des mots de passe en texte clair, soit ne pas utiliser du tout ces applications client/serveur sur votre réseau.

17.2. Terminologie Kerberos

Kerberos dispose de sa propre terminologie pour définir différents aspects du service. Avant d'évoquer la manière dont Kerberos fonctionne, il convient de se familiariser avec les termes suivants:

ciphertext

Données cryptées.

client

Entité sur le réseau (utilisateur, hôte ou application) pouvant obtenir un ticket Kerberos.

cache de certificat d'identité ou fichier de ticket

Fichier contenant les clés nécessaires au cryptage des communications entre un utilisateur et divers services réseau. Kerberos 5 fournit un environnement permettant d'utiliser d'autres types de cache (par exemple, une mémoire partagée), mais les fichiers sont mieux pris en charge de cette façon.

hache crypté

Hache unidirectionnel utilisé pour l'authentification des utilisateurs. Plus sûr que le texte clair, mais relativement facile à décoder pour un pirate expérimenté.

GSS-API

'Generic Security Service Application Program Interface' (GSS-API) [RFC-2743] est un ensemble de fonctions fournissant des services de sécurité. Les clients peuvent les utiliser pour leur authentification auprès des serveurs et les serveurs peuvent peuvent y avoir recours pour leur authentification auprès des clients sans devoir comprendre le mécanisme de fonctionnement sous-jacent. Si un service de réseau (comme IMPAP) utilise GSS-API, il peut se servir de Kerberos pour des besoins d'authentification.

clé

Bloc de données utilisé pour le cryptage et le décryptage de données. Il est impossible de décrypter des données cryptées sans disposer de la clé appropriée, à moins d'être un génie en devinettes.

Key Distribution Center (KDC)

Service émettant des tickets Kerberos, généralement exécuté sur le même hôte que le Serveur d'émission de tickets.

table clé ou keytab

Fichier contenant une liste cryptée des "principaux" et de leurs clés respectives. Les serveurs extraient les clés dont ils ont besoin des fichiers keytab au lieu d'utiliser `kinit`. Le fichier keytab par défaut est `/etc/krb5.keytab`. Le serveur d'administration de KDC, `/usr/kerberos/sbin/kadmind`, est le seul service utilisant tout autre fichier (il utilise `/var/kerberos/krb5kdc/kadm5.keytab`).

`kinit`

La commande `kinit` permet à un principal qui est déjà connecté d'obtenir et de mettre en cache le Ticket d'émission de tickets (TGT) initial. Pour de plus amples informations sur l'utilisation de la commande `kinit`, consultez sa page de manuel.

principal

Le principal est le nom unique de l'utilisateur ou du service pouvant effectuer une authentification à l'aide de Kerberos. Un nom de principal a la forme `root[/instance]@REALM`. Pour un utilisateur ordinaire, la variable root correspond à l'ID de connexion. L'`instance` est facultative. Si le principal a une instance, il est séparé de la variable root par une barre oblique en avant (/). Une chaîne vide ("") est considérée comme une instance valide (qui diffère de l'instance `NULL` par défaut), mais son utilisation peut être source de confusion. Tous les éléments principaux d'une zone (realm) ont leur propre clé dérivée de leur mot de passe ou définie de façon aléatoire pour les services.

realm

Un réseau utilisant Kerberos, composé d'un ou plusieurs serveurs (appelés également KDC) et un nombre potentiel très élevé de clients.

service

Programme accessible via le réseau.

ticket

Ensemble temporaire de certificats d'identité électroniques indiquant l'identité d'un client pour un service particulier.

Service d'émission de tickets ('Ticket Granting Service' ou TGS)

Serveur délivrant les tickets pour un service demandé que l'utilisateur doit ensuite employer pour accéder au service en question. TGS fonctionne en général sur le même hôte que KDC.

Ticket d'émission de tickets ('Ticket Granting Ticket' ou TGT)

Ticket spécial permettant au client d'obtenir des tickets supplémentaires sans les demander au KDC.

mot de passe non-crypté

Un mot de passe en texte clair, lisible par quiconque.

17.3. Fonctionnement de Kerberos

Kerberos est différent des autres méthodes d'authentification. Plutôt que de laisser l'authentification avoir lieu entre chaque machine cliente et chaque serveur, Kerberos utilise un cryptage symétrique et un programme fiable — connu sous le nom de Centre distributeur de tickets (KDC, Key Distribution Center) — afin d'authentifier les utilisateurs sur un réseau. Une fois l'authentification effectuée,

Kerberos stocke un ticket spécifique à cette session sur l'ordinateur de l'utilisateur et les services 'kerberisés' rechercheront ce ticket au lieu de demander à l'utilisateur de s'authentifier à l'aide d'un mot de passe.

Lorsqu'un utilisateur d'un réseau "kerberisé" se connecte sur son poste de travail, son principal est envoyé au KDC comme une demande de TGT. Cette demande peut être émise par le programme de connexion (de sorte qu'elle est transparente pour l'utilisateur) ou peut être émise par le programme kinit une fois l'utilisateur connecté.

Le KDC vérifie la présence du principal dans sa base de données. Si le principal y figure, le KDC crée un TGT, le crypte à l'aide de la clé de l'utilisateur, puis le renvoie à ce dernier.

Le programme de connexion ou le progamme kinit présent sur l'ordinateur client décrypte ensuite le TGT à l'aide de la clé de l'utilisateur (qu'il recompose à partir du mot de passe). La clé de l'utilisateur est utilisée seulement sur l'ordinateur client et *n'*est *pas* envoyée sur le réseau.

Le TGT, établi pour expirer après un certain laps de temps (généralement dix heures), est stocké dans un cache de certificats d'identité de l'ordinateur client. Un délai d'expiration est défini de manière à ce qu'un TGT compromis ne puisse être utilisé par un pirate que pendant une courte durée. Une fois que le TGT est émis, l'utilisateur n'a pas à redonner son mot de passe au KDC tant que le TGT n'a pas expiré ou tant qu'il ne se déconnecte pas pour se reconnecte ensuite.

Chaque fois que l'utilisateur doit accéder à un service réseau, le logiciel client utilise le TGT pour demander au TGS un nouveau ticket pour ce service spécifique. Le ticket pour le service souhaité est alors émis et utilisé pour authentifier l'utilisateur auprès de ce service de façon transparente.



Avertissement

Le système Kerberos peut être compromis à chaque fois qu'un utilisateur présent sur le réseau s'authentifie auprès d'un service "non-kerberisé" en envoyant un mot de passe en texte en clair. Pour cette raison, l'utilisation d'un service "non-kerberisé" est fortement déconseillée. Parmi de tels services figurent Telnet et FTP. L'utilisation d'autres protocoles sûrs, tels que les services sécurisés OpenSSH ou SSL, est certes acceptable mais pas idéale.

Ceci est bien sûr un aperçu général du fonctionnement typique de l'authentification de Kerberos sur un réseau. Pour obtenir des informations plus détaillées sur ce sujet, reportez-vous à la Section 17.7.



Remarque

Le bon fonctionnement de Kerberos dépend de certains services réseau. Il a tout d'abord besoin d'une synchronisation approximative de l'horloge entre les différents ordinateurs du réseau. Par conséquent, un programme de synchronisation de l'horloge devrait être installé pour le réseau, comme par exemple, ntpd. Pour de plus amples informations sur la configuration de ntpd, consultez /usr/share/doc/ntp-<*version-number*>/index.htm et examinez les renseignements concernant la configuration des serveur 'Network Time Protocol'.

En outre, étant donné que certains aspects de Kerberos reposent sur le DNS ('Domain Name Service'), assurez-vous que les entrées DNS et les hôtes sur le réseau soient tous correctement configurés. Pour plus d'informations, reportez-vous au *Guide de l'administrateur système Kerberos V5* disponible en formats PostScript et HTML dans /usr/share/doc/krb5-server-<*version-number*>.

17.4. Kerberos et PAM (modules d'authentification enfichables)

Actuellement, les services "kerberisés" n'utilisent pas du tout les PAM (Pluggable Authentication Modules) — les serveurs "kerberisés" ignorent complètement les PAM. Toutefois, les applications utilisant des PAM peuvent se servir de Kerberos pour l'authentification si le module `pam_krb5` (contenu dans le paquetage `pam_krb5`) est installé. Le paquetage `pam_krb5` contient des exemples de fichiers de configuration qui permettent à des services tels que `login` et `gdm` d'authentifier des utilisateurs et d'obtenir des certificats d'identité initiaux à l'aide de leurs mots de passe. Pour autant que l'accès aux serveurs de réseau s'effectue toujours à l'aide de services "kerberisés", ou de services utilisant GSS-API, par exemple IMAP, le réseau peut être considéré comme raisonnablement sûr.

Les administrateurs s'assureront de ne pas permettre l'authentification des utilisateur auprès de la plupart des réseaux au moyen de leurs mots de passe Kerberos. En effet, de nombreux protocoles utilisés par ces services ne cryptent pas le mot de passe avant de l'envoyer sur le réseau, annulant ainsi tous les avantages d'un système Korberos. Les utilisateurs ne devraient par exemple pas être autorisés à s'authentifier au moyen de leur mot de passe Kerberos sur un réseau Telnet.

La section suivante va décrire de quelle façon configurer un serveur Kerberos de base.

17.5. Configuration d'un serveur Kerberos 5

Lors de la configuration de Kerberos, installez tout d'abord le serveur. Si vous devez configurer des serveurs esclaves, les relations de configuration entre les serveurs maîtres et esclaves sont présentées de façon détaillée dans le *Guide d'installation Kerberos V5* (dans `/usr/share/doc/krb5-server-<numéro-version>`).

Pour installer un serveur Kerberos, suivez les étapes suivantes:

1. Avant d'installer Kerberos 5, assurez-vous que la synchronisation de l'horloge et que le DNS fonctionnent sur votre serveur. Prêtez une attention toute particulière à la synchronisation de l'heure entre le serveur Kerberos et ses différents clients. Si les horloges du serveur et du client diffèrent de plus de cinq minutes (cette durée par défaut est configurable dans Kerberos 5), les clients Kerberos ne pourront pas s'authentifier auprès du serveur. Cette synchronisation de l'horloge est nécessaire pour empêcher un pirate d'utiliser un ancien ticket pour se faire passer pour un utilisateur valide.

   Vous devriez configurer un réseau client/serveur compatible NTP (protocole de synchronisation de réseau) même si vous utilisez Kerberos. Afin de faciliter l'installation, Red Hat Linux inclut le paquetage `ntp`. Consultez `/usr/share/doc/ntp-<version-number>/index.htm` pour obtenir des informations détaillées sur la configuration des serveurs Network Time Protocol et rendez-vous à l'adresse suivante: http://www.eecis.udel.edu/~ntp pour obtenir des informations supplémentaires sur NTP.

2. Installez les paquetages `krb5-libs`, `krb5-server` et `krb5-workstation` sur la machine choisie pour l'exécution du KDC. Cette machine doit être absolument sécurisée — dans la mesure du possible, elle ne devrait exécuter aucun service autre que le KDC.

   Si vous souhaitez utiliser un utilitaire d'interface utilisateur graphique (GUI) pour l'administration de Kerberos, vous devez également installer le paquetage `gnome-kerberos`. Celui-ci contient `krb5`, un outil GUI pour gérer les tickets.

3. Éditez les fichiers de configuration `/etc/krb5.conf` et `/var/kerberos/krb5kdc/kdc.conf` afin qu'ils correspondent à vos mappages nom du realm et domaine-à-realm. Un simple realm peut être construit en remplaçant des instances de *`EXAMPLE.COM`* et *`example.com`* par votre nom de domaine — en vous assurant de bien respecter le format correct des noms contenant des lettres majuscules et minuscules — et en changeant le KDC dans *`kerberos.example.com`* pour le nom de votre serveur Kerberos. Par convention, tous les noms de realm sont en lettres majuscules et tous les noms de d'hôtes et

de domaines DNS sont en lettres minuscules. Pour des informations détaillées sur le format de ces fichiers, consultez leur page de manuel respectives.

4. Créez la base de données en utilisant l'utilitaire `kdb5_util` à partir de l'invite du shell:

   ```
   /usr/kerberos/sbin/kdb5_util create -s
   ```

   La commande `create` crée la base de données qui sera utilisée pour stocker les clés pour votre realm dans Kerberos. L'option `-s` permet la création forcée d'un fichier *stash* dans lequel la clé du serveurs maître est stockée. En l'absence d'un fichier stash à partir duquel la clé peut être lue, le serveur Korberos (`krb5kdc`) enverra une invite pour que l'utilisateur entre le mot de passe du serveur maître (qui permet de recréer la clé) à chaque fois qu'il sera lancé.

5. Éditez le fichier `/var/kerberos/krb5kdc/kadm5.acl` Ce fichier est utilisé par `kadmind` afin de déterminer d'une part quels éléments principaux ont un accès administratif à la base de données de Kerberos et d'autre part, afin de définir leur niveau d'accès. Une seule ligne suffira à la plupart des organisations, comme dans l'exemple ci-dessous:

   ```
   */admin@EXAMPLE.COM  *
   ```

   La plupart des utilisateurs seront représentés dans la base de données par un seul élément principal (avec une instance *NULL*, ou vide, telle que *joe@EXAMPLE.COM*). Avec cette configuration, les utilisateurs ayant un second élément principal avec comme instance *admin* (par exemple, *joe/admin@EXAMPLE.COM*) pourront exercer un pouvoir total sur la base de données Kerberos du realm.

   Une fois que `kadmind` est lancé sur le serveur, tout utilisateur pourra accéder à ses services en exécutant `kadmin` sur tout client ou serveurs dans le realm. Toutefois, les utilisateurs non-spécifiés dans le fichier `kadm5.acl` ne pourront modifier le contenu de la base de données d'aucune manière, à l'exception de leurs propres mots de passe qu'ils seront à même de changer.

   Remarque

   L'utilitaire `kadmin` communique avec le serveur `kadmind` sur le réseau et utilise Kerberos pour gérer l'authentification. Bien sûr, vous devez créer le premier élément principal avant de pouvoir vous connecter au serveur sur le réseau afin qu'il puisse le gérer. Pour créer le premier élément principal, utilisez `kadmin.local`, une commande conçue spécifiquement pour être utilisée sur le même hôte que le KDC et qui n'emploie pas Kerberos pour l'authentification.

   Tapez la commande `kadmin.local` suivante sur terminal KDC afin de créer le premier élément principal:

   ```
   /usr/kerberos/sbin/kadmin.local -q "addprinc
   username/admin"
   ```

6. Lancez Kerberos à l'aide des commandes suivantes:

   ```
   /sbin/service krb5kdc start
   /sbin/service kadmin start
   /sbin/service krb524 start
   ```

7. Ajoutez des éléments principaux pour vos utilisateurs à l'aide de la commande `addprinc` avec `kadmin`. Les commandes `kadmin` et `kadmin.local` sont des interfaces de ligne de commande vers le KDC. En tant que telles, de nombreuses commandes sont disponibles après le lancement du programme `kadmin`. Veuillez vous référez à la page de manuel relative à `kadmin` pour plus d'informations.

8. Vérifiez que votre serveur émettra bien des tickets. Tout d'abord, exécutez `kinit` afin d'obtenir un ticket et de le stocker dans un fichier de cache de certificats d'identité. Utilisez ensuite `klist` pour visualiser la liste des certificats d'identité dans votre cache et utilisez `kdestroy` pour détruire le cache et les certificats qu'il contient.



Remarque

Par défaut, `kinit` tente de vous authentifier à l'aide du nom d'utilisateur de connexion associé au compte utilisé lorsque vous vous êtes connecté pour la première fois à votre système (pas au serveur Kerberos). Si le nom d'utilisateur de ce système ne correspond pas à un élément principal dans votre base de données Korberos, un message d'erreur s'affichera. Dans ce cas, indiquez simplement à `kinit` le nom de votre élément principal en tant qu'argument sur la ligne de commande (`kinit` *`élément principal`*).

Une fois les étapes ci-dessus réalisées, votre serveur Kerberos devrait être opérationnel. Vous devrez ensuite configurer vos clients Kerberos.

17.6. Configurer un client Kerberos 5

Il est moins complexe de configurer un client Kerberos 5 qu'un serveur. Vous devez au minimum installer les paquetages clients et fournir à vos clients un fichier de configuration `krb5.conf` valide. Les versions "kerberisées" de `rsh` et `rlogin` devront également être modifiée au niveau de la configuration.

1. Assurez-vous que la synchronisation de l'heure est bien établie entre le client Kerberos et le KDC. Reportez-vous à la Section 17.5 pour de plus amples informations. En outre, vérifiez que le DNS fonctionne correctement sur le client Kerberos avant d'installer les programmes de ce client.
2. Installez les paquetages `krb5-libs` et `krb5-workstation` sur tous les clients de votre realm. Vous devez fournir une version de `/etc/krb5.conf` pour chacun de vos clients; généralement, le fichier `krb5.conf` utilisé pour le KDC peut également servir ici.
3. Avant qu'une station de travail spécifiée dans le realm puisse permettre aux utilisateurs de se connecter à l'aide des commandes "kerberisées" `rsh` et `rlogin`, le paquetage `xinetd` devra y être installé et l'élément principal de l'hôte propre à la station devra être présent dans la base de données Kerberos. Les programmes de serveur `kshd` et `klogind` auront également besoin d'un accès aux clés pour l'élément principal de leur service.

   À l'aide de `kadmin`, ajoutez un élément principal d'hôte pour la station de travail sur le KDC. L'instance sera dans ce cas le nom d'hôte de la station de travail. Vous pouvez utiliser l'option `-randkey` de la commande `addprinc` de `kadmin` pour créer l'élément principal et lui attribuer une clé aléatoire:

   ```
   addprinc -randkey
   host/blah.example.com
   ```

   Maintenant que vous avez créé l'élément principal, vous pouvez extraire les clés de la station de travail en exécutant `kadmin` *sur la station de travail elle-même*, et en utilisant la commande `ktadd` dans `kadmin`:

   ```
   ktadd -k /etc/krb5.keytab
   host/blah.example.com
   ```

4. Si vous souhaitez utiliser d'autres services réseau "kerberisés", vous devrez les démarrer. Ci-dessous figure une liste des services "kerberisés" les plus courants et les instructions relatives à leur activation:
   - `rsh` and `rlogin` — Afin d'utiliser les versions "kerberisées" de `rsh` et `rlogin`, vous devez activer `klogin`, `eklogin`, et `kshell`.
   - Telnet — Afin d'utiliser le service "kerberisé" Telnet, vous devez activer `krb5-telnet`.
   - FTP — Afin de fournir un accès FTP, créez puis extrayez une clé pour un élément principal avec un root de `ftp`. Pour cette opération l'instance doit être configurée au nom d'hôte du serveur FTP. Activez ensuite `gssftp`.

- IMAP — Le serveur IMAP inclus dans le paquetage `imap` utilisera l'authentification GSS-API à l'aide de Kerberos 5 s'il parvient à trouver la clé appropriée dans `/etc/krb5.keytab`. Le root de l'élément principal devrait être `imap`.
- CVS — Le `gserver` "kerberisé" de CVS utilise un élément principal avec un root de `cvs` et, hormis ce point, est identique au `pserver` de CVS.

Reportez-vous au chapitre intitulé *Contrôle de l'accès aux services* dans le *Guide de personnalisation de Red Hat Linux* pour obtenir de plus amples informations sur l'activation des services.

17.7. Ressources supplémentaires

Pour plus d'informations sur Kerberos, reportez-vous aux sources d'informations suivantes.

17.7.1. Documentation installée

- `/usr/share/doc/krb5-server-`<*version-number*> — Le *Guide d'installation Kerberos V5* et le *Guide de l'administrateur système Kerberos V5* dans des formats PostScript et HTML. Le paquetage `krb5-server` doit être installé.
- `/usr/share/doc/krb5-workstation-`<*version-number*> — Le *Guide de l'utilisateur Kerberos V5 UNIX* dans des formats PostScript et HTML. Le paquetage `krb5-workstation` doit être installé.

17.7.2. Sites Web utiles

- http://web.mit.edu/kerberos/www — La page *Kerberos: The Network Authentication Protocol* (Kerberos: le protocole d'authentification réseau) sur le site Web du MIT.
- http://www.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html — Le Forum Aux Questions (FAQ) de Kerberos.
- ftp://athena-dist.mit.edu/pub/kerberos/doc/usenix.PS — Lien vers la version PostScript de *Kerberos: An Authentication Service for Open Network Systems* (Kerberos: un service d'authentification pour des systèmes de réseau ouvert) par Jennifer G. Steiner, Clifford Neuman et Jeffrey I. Schiller. Il s'agit du document original décrivant Kerberos.
- http://web.mit.edu/kerberos/www/dialogue.html — *Designing an Authentication System: a Dialogue in Four Scenes* (Conception d'un système d'authentification: un dialogue en quatre parties) écrit par Bill Bryant en 1988, puis modifié par Theodore Ts'o en 1997. Ce document relate une conversation entre deux développeurs réfléchissant à la création d'un système d'authentification de type Kerberos. La présentation sous forme de dialogue en font un bon point de départ pour les néophytes.
- http://www.ornl.gov/~jar/HowToKerb.html — *How to Kerberize your site* (Comment "kerbébériser" votre site) est une excellente référence pour la "kerberisation" d'un réseau.
- http://www.networkcomputing.com/netdesign/kerb1.html — *Kerberos Network Design Manual* (Manuel pour la conception d'un réseau Kerberos) offre un aperçu complet du système Kerberos.



Chapitre 18.

Protocole SSH

SSH™ permet aux utilisateurs de se connecter à distance à d'autres systèmes hôtes. Contrairement à des protocoles tels que FTP ou Telnet, SSH crypte session de connexion et empêche ainsi tout agresseurs d'amasser des mots de passe en texte clair.

SSH est conçu pour remplacer les applications de terminal plus anciennes et moins sécurisées au moyen desquelles il est possible de se connecter à des hôtes distants, comme **telnet** ou **rsh**. Un programme similaire appelé **scp** remplace des programmes moins récents conçus pour copier les fichiers entre les différents hôtes, tels que **rcp**. Étant donné que ces applications ne cryptent pas les mots de passe entre le client et le serveur, il est recommandé d'éviter autant que possible leur utilisation. En ayant recours à des méthodes sécurisées pour vous connecter à distance à d'autres systèmes, les risques en matière de sécurité, aussi bien pour le système client que pour l'hôte distant sont considérablement réduits.

18.1. Fonctionnalités de SSH

SSH (ou *S*ecure *SH*ell) est un protocole servant à créer une connexion sécurisée entre deux systèmes utilisant une architecture serveur/client.

SSH offre les garanties de sécurité suivantes:

- Après avoir effectué une connexion initiale, le client peut s'assurer que sa connexion est établie avec le même serveur que lors de sa session précédente.
- Le client transmet ses données d'authentification au serveur au moyen d'un cryptage 128 bits.
- Toutes les données envoyées et reçues lors d'une session sont transférés au moyen d'un cryptage 128 bits, rendant par là-même le décryptage et la lecture de toute transmission interceptée extrêmement difficile.
- Le client a la possibilité d'utiliser des applications X11 [1] à partir du serveur. Cette technique, appelée *retransmission X11*, fournit un moyen sécurisé pour l'utilisation d'applications graphiques sur un réseau.

Étant donné que le protocole SSH crypte tout ce qu'il envoie et reçoit, il peut être utilisé pour sécuriser des protocoles non-sûrs. Grâce à la technique de *retransmission de port*, un serveur SSH peut être employé pour sécuriser des protocoles non-sûrs, tel que POP, augmentant ainsi la sécurité du système et de ses données.

Red Hat Linux contient le paquetage OpenSSH général, (`openssh`), les paquetages serveur OpenSSH (`openssh-server`) et client OpenSSH (`openssh-clients`). Veuillez consultez le chapitre intitulé *OpenSSH* du *Guide de personnalisation de Red Hat Linux* pour obtenir des instructions d'installation et d'utilisation d'OpenSSH. Notez également que les paquetages OpenSSH ont besoin du paquetage OpenSSL (`openssl`). OpenSSL installe de nombreuses bibliothèques cryptographiques importantes qui permettent à OpenSSH de crypter les communications.

Un grand nombre de programmes client et serveur peuvent utiliser le protocole SSH. Il existe des applications SSH clients pour presque tous les systèmes d'exploitation utilisés à l'heure actuelle.

1. X11 fait référence au système d'affichage de fenêtres X11R6, généralement appelé X. Red Hat Linux comprend **XFree86**, un système X Window Open Source très utilisé, basé sur X11R6.

18.1.1. Pourquoi utiliser SSH?

Les dangereux utilisateurs d'ordinateurs disposent d'une variété d'outils pour désorganiser, intercepter et réacheminer le trafic de réseaux et forcer l'accès à votre système. De manière générale, ces menaces peuvent être répertoriées comme suit:

- *Interception d'une communication entre deux systèmes* — Ce scénario implique que le pirate peut se trouver n'importe où sur le réseau entre les entités qui communiquement, et peut copier les informations qu'elles se transmettent. Le pirate peut copier et garder les informations ou peut les modifier avant de les envoyer au destinataire prévu.

  Cette attaque peut être orchestrée en utilisant un programme renifleur — un utilitaire réseau courant.

- *Usurpation de l'identité d'un hôte* — grâce à cette technique, un système intercepteur prétend être le destinataire désiré d'un message. Si cet abus d'identité fonctionne, le client ne s'en rend pas compte et continue de lui envoyer toutes les informations, comme s'il était connecté au bon destinataire.

  Ce type d'attaque peut être organisé grâce à l'utilisation de techniques connues sous le nom d'empoisonnements DNS [2] ou usurpation d'adresse IP [3].

Ces deux techniques interceptent potentiellement des informations confidentielles et si cette interception est effectuée pour des raisons hostiles, le résultat peut être catastrophique.

L'utilisation du protocole SSH pour effectuer une connexion shell à distance ou copier des fichiers permet de faire réduire considérablement ces atteintes à la sécurité. La signature numérique d'un serveur fournit la vérification pour son identité. En outre, la communication complète entre un système client et un système serveur ne peut être utilisée si elle est interceptée, car tous les paquets sont cryptés. De plus, il n'est pas possible d'usurper l'identité d'un des deux systèmes, parce que les paquets sont cryptés et leurs clés ne sont connues que par le système local et le système distant.

18.2. Versions du protocole SSH

Le protocole SSH permet à tout programme client et serveur créé selon les spécifications du protocole, de communiquer de façon sécurisée et d'être utilisé de manière interchangeable.

À l'heure actuelle, il existe deux types différents de protocoles SSH. La version 1 contient de nombreux algorithmes de cryptage brevetés (toutefois, bon nombre de ces brevets sont périmés) et expose des brèches sécurité qui donnent la possibilité éventuelle à un agresseur d'insérer des données dans le flux de communication. Sous Red Hat Linux, la suite OpenSSH suite utilise par défaut la version SSH 2.0, bien qu'elle prenne en charge la version 1.



Important

Il est conseillé de n'utiliser, autant que possible, des serveurs et clients compatibles avec la version 2.

2. L'empoisonnement DNS a lieu lorsqu'un intrus pénètre sur un serveur DNS, dirigeant les systèmes client vers un hôte double avec une intention malveillante.
3. L'usurpation d'adresse IP se produit lorsqu'un intrus envoie des paquets réseau qui apparaissent faussement sur le réseau, comme provenant d'un hôte de confiance.

18.3. Séquence des événements d'une connexion SSH

Pour aider à protéger l'intégrité d'une communication SSH entre deux ordinateurs hôtes, la série suivante d'événements doit être utilisée.

- Une liaison cryptographique est établie afin de permettre au client de vérifier qu'il est bien en communication avec le serveur souhaité.
- La couche transport de la connexion entre le client et un hôte distant est cryptée au moyen d'un chiffre symétrique.
- Le client s'authentifie auprès du serveur.
- Le client distant peut désormais interagir de manière sécurisée avec l'hôte distant au moyen d'une connexion cryptée.

18.3.1. Couche transport

Le rôle principal d'une couche transport est de faciliter une communication sécurisée entre deux hôtes non seulement au moment de l'authentification, mais également après. Pour ce faire, la couche transport traite le cryptage et décryptage de données et offre la protection de l'intégrité des paquets de données lors de leur envoi et de leur réception. De plus, la couche transport effectue la compression des données permettant l'accélération la vitesse de transfert d'information.

Lorsqu'un client communique avec un serveur au moyen d'un protocole SSH, de nombreux éléments importants sont négociés afin que les deux systèmes puissent créer correctement la couche transport. Les opérations ci-dessous ont lieu durant cet échange:

- des clés sont échangées;
- l'algorithme de cryptage de clés publiques est déterminé;
- l'algorithme de cryptage symétrique est déterminé;
- l'algorithme d'authentification de message est déterminé;
- l'algorithme de hachage est déterminé.

Durant l'échange des clés, le serveur s'identifie au client au moyen d'une *clé d'hôte* unique. Évidemment, si le client communique pour la première fois avec ce serveur, la clé du serveur ne sera pas connue du client et la connexion ne pourra être établie. OpenSSH contourne ce problème en acceptant la clé d'hôte du serveur après notification de l'utilisateur et vérifie l'acceptation de la nouvelle clé d'hôte. Lors des connexions suivantes, la clé d'hôte du serveur peut être vérifiée au moyen d'une version enregistrée sur le client, permettant ainsi au client de s'assurer qu'il communique bien avec le serveur désiré. Si, à l'avenir, la clé d'hôte n'est plus valide, l'utilisateur doit supprimer la version sauvegardée du client avant qu'une nouvelle connexion ne puisse avoir lieu.



Attention

Un pirate pourrait se faire passer pour le serveur SSH lors de la première connexion car le système local ne reconnaît aucune différence entre le serveur désiré et celui établit par le pirate. Afin d'éviter une telle situation, contrôlez l'intégrité d'un nouveau serveur SSH en contactant l'administrateur du serveur avant d'établir la première connexion ou dans le cas d'une clé d'hôte sans correspondance valide.

Le protocole SSH est conçu pour fonctionner avec la plupart des types d'algorithme de clé publique ou de format de codage. Après la création de deux valeurs lors de l'échange initial des clés (une valeur de

hachage utilisée pour les échanges et une valeur secrète partagée), les deux systèmes commencent immédiatement à calculer de nouveaux algorithmes et de nouvelles clés pour protéger l'authentification et les données qui seront envoyées au cours de la connexion.

Après qu'une certaine quantité de données a été transmise au moyen d'une clé et d'un algorithme précis (la quantité exacte dépend de la mise en application du protocole SSH), un nouvel échange de clés s'effectue et produit un autre ensemble de valeurs de hachage et une autre valeur secrète partagée. De cette façon, même si un pirate réussit à déterminer les valeurs de hachage et la valeur secrète partagée, ces informations ne lui seront utiles que pour une durée limitée.

18.3.2. Authentification

Une fois que la couche transport a créé un tunnel sécurisé pour envoyer les informations entre les deux systèmes, le serveur indique au client les différentes méthodes d'authentification prises en charge, telles que l'utilisation d'une signature dotée d'une clé codée ou la saisie d'un mot de passe. Le client doit ensuite essayer de s'authentifier auprès du serveur au moyen d'une des méthodes spécifiées.

Les serveurs et clients SSH pouvant être configurés de façon à permettre différents types d'authentification, chacune des deux parties se voit attribuer un niveau de contrôle optimal. Le serveur peut décider des méthodes de cryptage à prendre en charge en fonction de son modèle de sécurité et le client peut choisir l'ordre des méthodes d'authentification à utiliser parmi les options disponibles. Grâce à la nature sécurisée de la couche transport SSH, même les méthodes d'authentification qui, au premier abord semblent non-sécurisées, telles que l'authentification basée sur l'hôte et le mot de passe, peuvent être utilisées en toute sécurité.

18.3.3. Canaux

Après avoir effectué avec succès l'authentification au moyen de la couche transport SSH, des *canaux* multiples sont ouverts au moyen d'une technique appelée multiplexage[4]. Chacun de ces canaux peut ainsi s'occuper de la communication de sessions de terminal différentes d'une part et des sessions de retransmission X11 d'autre part.

Le client et le serveur peuvent tous deux créer un nouveau canal. Chaque canal reçoit ensuite un numéro différent à chaque extrémité de la connexion. Lorsque le client essaie d'ouvrir un nouveau canal, il envoie le numéro du canal accompagné de la requête. Cette information est stockée par le serveur et utilisée pour adresser la communication à ce canal. Cette procédure est utilisée afin que des types différents de session ne créent des nuisances mutuelles et afin que, à la fin d'une session donnée, son canal puisse être fermé sans que la connexion SSH primaire ne soit interrompue.

Les canaux prennent aussi en charge le *contrôle du flux de données*, ce qui leur permet d'envoyer et de recevoir des données de façon ordonnée. Ce faisant, aucune donnée n'est envoyée par le canal tant que l'hôte n'a pas reçu un message lui indiquant que le canal est ouvert.

Le client et le serveur négocient automatiquement la configuration de chaque canal, selon le type de service demandé par le client et le mode de connexion de l'utilisateur au réseau. Ceci permet de gérer facilement différents types de connexions distantes sans devoir changer l'infrastructure de base du protocole.

18.4. Fichiers de configuration d'OpenSSH

OpenSSH est constitué de deux ensembles de fichiers de configuration, un pour les programmes client (`ssh`, `scp` et `sftp`) et l'autre pour le service (`sshd`).

4. Une connexion multiplexe se compose de plusieurs signaux envoyés sur un support commun et partagé. Avec le protocole SSH, divers canaux sont envoyés sur une connexion sécurisée commune.

Les informations de configuration SSH qui s'appliquent à l'ensemble du système sont stockées dans le répertoire `/etc/ssh`:

- `moduli` — contient les groupes Diffie-Hellman utilisés pour l'échange de clés Diffie-Hellman qui est crucial pour la création d'une couche transport sécurisée. Lorsque les clés sont échangées au début d'une session SSH, une valeur secrète partagée ne pouvant être déterminée que conjointement par les deux parties est créée. Cette valeur est ensuite utilisée pour accorder l'authentification d'hôte.
- `ssh_config` — fichier de configuration client SSH pour l'ensemble du système. Il est écrasé si un même fichier est présent dans le répertoire personnel de l'utilisateur (`~/.ssh/config`).
- `sshd_config` — fichier de configuration pour le démon `sshd`.
- `ssh_host_dsa_key` — clé DSA privée utilisée par le démon `sshd`.
- `ssh_host_dsa_key.pub` — clé DSA publique utilisée par le démon `sshd`.
- `ssh_host_key` — clé RSA privée utilisée par le démon `sshd` pour la version 1 du protocole SSH.
- `ssh_host_key.pub` — clé RSA publique utilisée par le démon `sshd` pour la version 1 du protocole SSH.
- `ssh_host_rsa_key` — clé RSA privée utilisée par le démon `sshd` pour la version 2 du protocole SSH.
- `ssh_host_rsa_key.pub` — clé RSA publique utilisée par le démon `sshd` pour la version 2 du protocole SSH.

Les informations de configuration SSH spécifiques à l'utilisateur sont stockées dans son répertoire personnel à l'intérieur du répertoire `~/.ssh/`:

- `authorized_keys` — ce fichier contient une liste de clés publiques autorisées pour les serveurs. Lorsque le client se connecte à un serveur, ce dernier authentifie le client en vérifiant sa clé publique signée qui est stockée dans ce fichier.
- `id_dsa` — contient la clé DSA privée de l'utilisateur.
- `id_dsa.pub` — la clé DSA publique de l'utilisateur.
- `id_rsa` — la clé RSA privée utilisée par `ssh` pour la version 2 du protocole SSH.
- `id_rsa.pub` — la clé RSA publique utilisée par `ssh` pour la version 2 du protocole SSH.
- `identity` — la clé RSA privée utilisée par `ssh` pour la version 1 du protocole SSH.
- `identity.pub` — la clé RSA privée utilisée par `ssh` pour la version 1 du protocole SSH.
- `known_hosts` — ce fichier contient les clés d'hôtes DSA des serveurs SSH auxquels l'utilisateur a eu accès. Ce fichier est très important car il permet de garantir que le client SSH se connecte au bon serveur SSH.



Important

Si la clé d'hôte d'un serveur SSH a changé, le client informera l'utilisateur que le processus de connexion ne peut se poursuivre tant que que la clé d'hôte du serveur n'a pas été supprimée du fichier `known_hosts` en utilisant un éditeur de texte. Avant de procéder à cette opération, il est conseillé de contacter l'administrateur système du serveur SSH pour vous assurer que le serveur n'est pas compromis.

Veuillez lire les pages de manuel concernant `ssh` et `sshd` pour avoir plus de détails sur les différentes directives disponibles dans les fichiers de configuration SSH.

18.5. Beaucoup plus qu'un shell sécurisé

Une interface sécurisée en ligne de commande n'est que la première utilisation, parmi tant d'autres, de SSH. En ayant la quantité nécessaire de bande passante, les sessions X11 peuvent être dirigées sur un canal SSH ou bien, en utilisant la retransmission TCP/IP, les connexions par port entre systèmes, considérées auparavant comme étant non-sécurisés, peuvent être appliquées à des canaux SSH spécifiques.

18.5.1. Retransmission X11

L'ouverture d'une session X11 par le biais d'une connexion SSH établie est aussi facile que l'exécution d'un programme X sur un ordinateur local. Lorsqu'un programme X est exécuté à partir d'un invite du shell sécurisée, le client et le serveur SSH créent un nouveau canal sécurisé et les données du programme X sont ensuite envoyées à l'ordinateur client par ce canal de façon transparente.

La retransmission X11 peut être très utile. Elle peut être utilisée par exemple, pour créer une session interactive sécurisée avec `up2date`. Pour ce faire, connectez-vous au serveur en utilisant **ssh** et en tapant:

```
up2date &
```

Après avoir fourni le mot de passe super-utilisateur pour le serveur, l'**Agent de mise à jour Red Hat** apparaîtra et permettra à l'utilisateur distant de mettre à jour en toute sécurité son système distant.

18.5.2. Retransmission de port

Grâce à SSH, il est possible de sécuriser des protocoles TCP/IP non-sécurisés via la retransmission de port. En utilisant cette technique, le serveur SSH devient un conduit crypté vers le client SSH.

La retransmission de port consiste à mapper un port local du client vers un port distant du serveur. SSH permet de mapper tout port du serveur vers tout port du client, sans nécessité une correspondance des numéros de port pour un bon fonctionnement.

Pour créer un canal de retransmission de port TCP/IP qui attend les connexions sur l'hôte local, utilisez la commande suivante:

```
ssh -L port-local:hôte-distant:port-distant nom-d'utilisateur@nom-d'hôte
```



Remarque

Afin de pouvoir définir la retransmission de port pour qu'elle puisse être en mode réception des ports inférieurs à 1024, il est nécessaire d'avoir un accès super-utilisateur (ou root).

Pour vérifier le courrier électronique sur un serveur nommé mail.example.com au moyen du protocole POP à travers une connexion cryptée, utilisez la commande ci-dessous:

```
ssh -L 1100:mail.example.com:110 mail.example.com
```

Une fois que le canal de retransmission de port est en place entre l'ordinateur client et le serveur de courrier, dirigez le client POP mail pour qu'il utilise le port 1100 sur l'hôte local afin de vérifier le nouveau courrier. Toute requête envoyée au port 1100 de le système client sera dirigée de façon sécurisée vers le serveur mail.example.com.

Si mail.example.com n'exécute pas un serveur SSH, mais qu'un autre ordinateur le fait, SSH peut toujours être utilisé pour sécuriser une partie de la connexion. Dans ce cas, un commande légèrement différente est nécessaire:

```
ssh -L 1100:mail.example.com:110 other.example.com
```

Dans cet exemple, des requêtes POP du port 1100 sur l'ordinateur client sont transférées au moyen de la connexion SSH au port 22 vers le serveur SSH, `other.example.com`. Ensuite, `other.example.com` se connecte au port 110 de `mail.example.com` vérifier l'arrivée de nouveau courrier. Notez qu'en utilisant cette technique, seule la connexion entre le système client et le serveur SSH `other.example.com` est sécurisée.

La retransmission de ports peut être également utilisée pour obtenir des informations de façon sécurisée à travers un pare-feu. Si le pare-feu est configuré de façon à permettre le trafic SSH par son port standard (22), mais bloque l'accès aux autres ports, une connexion entre deux ordinateurs hôtes qui utilisent des ports bloqués est tout de même possible en redirigeant leur communication sur une connexion SSH établie entre eux.



Remarque

L'utilisation de la retransmission de port pour transférer des connexions de cette façon permet à tout utilisateur sur le système client de se connecter à ce service. Si le système client est compromis, les pirates auront également accès aux services retransmis.

Les administrateurs système inquiets quant à l'utilisation de la retransmission de port peuvent désactiver cette fonction sur le serveur en spécifiant le paramètre `No` pour la ligne AllowTcpForwarding dans `/etc/ssh/sshd_config` et ensuite redémarrer le service `sshd`.

18.6. Exiger SSH pour les connexions à distance

Afin que le protocole SSH soit vraiment efficace, il est essentiel de n'utiliser aucun protocole de connexion non-sécurisés, tels que Telnet et FTP. Autrement, le mot de passe d'un utilisateur sera certes peut-être protégé au moyen de SSH pour une session, mais il pourra être capté lors d'une connexion ultérieure au moyen de Telnet.

Ci-dessous figurent certains services que vous devez désactiver:

- `telnet`
- `rsh`
- `rlogin`
- `vsftpd`

Pour désactiver des méthodes de connexion non-sécurisées au système, utilisez le programme à ligne de commande `chkconfig`, le programme basé sur ncurses **ntsysv** ou l'application graphique **Outil de configuration des services** (`redhat-config-services`). Tous ces outils nécessitent un accès super-utilisateur (ou root).

Pour plus d'informations sur les niveaux d'exécution et la configuration des services à l'aide de `chkconfig`, `ntsysv` et **serviceconf**, consultez le chapitre intitulé *Contrôle de l'accès aux services* du *Guide de personnalisation de Red Hat Linux*.



Chapitre 19. Tripwire

Le logiciel Tripwire aide à assurer l'intégrité de répertoires et de systèmes de fichiers importants en identifiant tout changement apporté à ceux-ci. Pour ce faire, il effectue des vérifications automatiques à intervalle régulier. Si Tripwire détecte qu'un fichier sous surveillance a été modifié, il en informe l'administrateur système par courrier électronique. Parce que Tripwire peut identifier de façon positive les fichiers qui ont été ajoutés, modifiés ou supprimés, il peut accélérer la restauration de fichiers après une violation en réduisant au minimum le nombre de fichiers qui doivent être remis en état. Grâce à ces capacités, Tripwire est un excellent outil pour les administrateur système à la recherche d'un outil de détection des violations et d'évaluation des dégâts survenus sur leurs serveurs.

Tripwire compare des fichiers et des répertoires avec des informations, telles que des emplacements de fichier, des dates de modification de fichier et d'autres données. La base de données contient des *baselines* — qui sont des instantanés de répertoires et de fichiers spécifiques à un moment donné. Le contenu de la base de données référentielle doit être créée avant que le système ne coure le risque d'être victime d'une intrusion. Une fois la base de données référentielle créée, Tripwire compare le système en cours avec cette base de données et produit un rapport des modifications, des ajouts et des suppressions effectués.

Bien qu'étant un outil valide qui permet de contrôler l'état de sécurité de votre système, Tripwire n'est pas pris en charge par Red Hat, Inc. Pour d'avantage d'informations, reportez-vous au site Internet du projet Tripwire à l'adresse suivante: http://www.tripwire.org.

19.1. Comment utiliser Tripwire

L'organigramme suivant illustre l'utilisation de Tripwire:

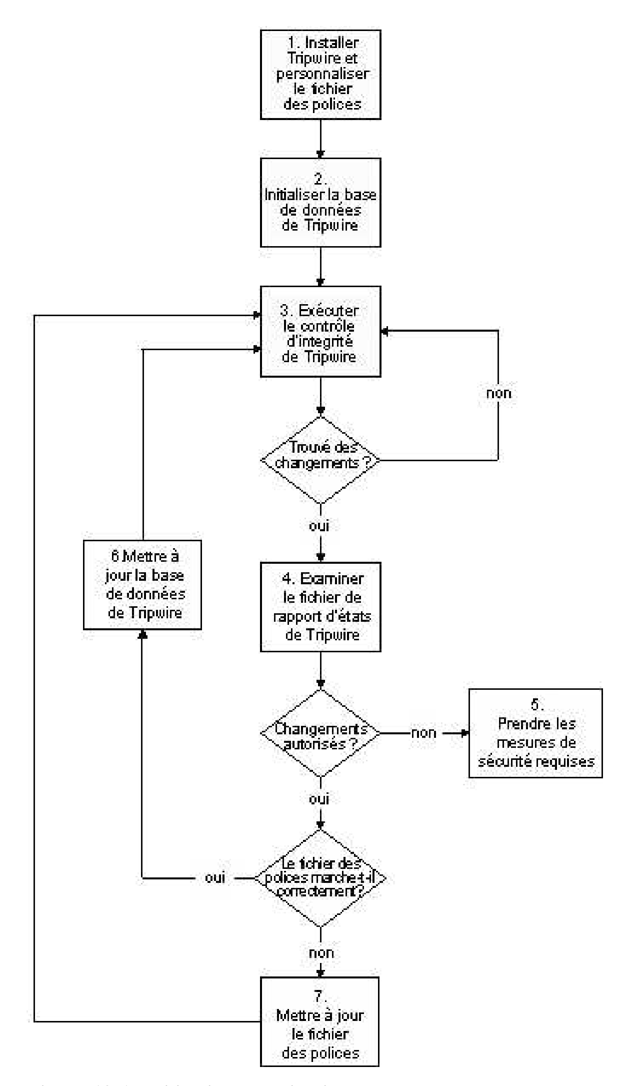


Figure 19-1. Utilisation de Tripwire

Les éléments suivants décrivent de façon détaillée les blocs illustrés dans la Figure 19-1.

1. Installation de Tripwire et personnalisation du fichier de politiques.

Installez le RPM de Tripwire (consultez la Section 19.2). Ensuite, personnalisez les exemples de fichiers de configuration et de politiques (à savoir `/etc/tripwire/twcfg.txt` et `/etc/tripwire/twpol.txt`), et exécutez le script de configuration `/etc/tripwire/twinstall.sh`. Pour de plus amples informations, reportez-vous à la Section 19.3.

2. Initialisation de la base de données de Tripwire

Créez une base de données des fichiers système critiques devant être contrôlés en fonction des directives contenues dans le tout nouveau fichier de politiques Tripwire signé, `/etc/tripwire/tw.pol`. Pour de plus amples informations, reportez-vous à la Section 19.4.

3. Exécution d'une vérification d'intégrité Tripwire

Comparez la base de données de Tripwire nouvellement créée avec les fichiers système pour vérifier s'il en manque ou si certains d'entre eux ont été modifiés. Reportez-vous à la Section 19.5.

4. Analyse d'un fichier rapport de Tripwire

Visualisez le fichier rapport Tripwire au moyen de `/usr/sbin/twprint` afin d'identifier les violations d'intégrité du système. Pour en savoir plus, reportez-vous à la Section 19.6.1.

5. Si des violations d'intégrité surviennent, prenez les mesures de sécurité appropriées

les fichiers contrôlés ont été modifiés de façon incorrecte, vous pouvez remplacer les fichiers originaux par des copies de sauvegarde, réinstaller le programme ou réinstaller complètement le système d'exploitation.

6. Si les modifications étaient valides, vérifiez et mettez à jour le fichier de la base de données de Tripwire.

Si les modifications de l'intégrité du système sont intentionnelles, vous devez indiquer au fichier de la base de données Tripwire de ne plus souligner ces modifications dans les rapports suivants. Pour plus de détails, veuillez lire la Section 19.7.

7. Si le dossier de politiques échappe à la vérification, mettez à jour le fichier de politiques Tripwire.

Pour changer la liste des fichiers que Tripwire contrôle ou la façon dont il traite ces violations d'intégrité, mettez à jour le fichier de politiques fourni (`/etc/tripwire/twpol.txt`), régénérez une copie signée (`/etc/tripwire/tw.pol`), et mettez à jour la base de données Tripwire. Pour plus de renseignements, reportez-vous à la Section 19.8.

Pour obtenir des instructions plus détaillées sur ces différentes étapes, consultez les sections de ce chapitre les concernant.

19.2. Installation du RPM de Tripwire

La façon la plus simple d'installer Tripwire consiste à sélectionner le RPM de Tripwire lors du processus d'installation Red Hat Linux. Toutefois, si Red Hat Linux est déjà installé, vous pouvez utiliser la commande `rpm` ou l' **Outil de gestion de paquetages** (`redhat-config-packages`) pour installer le RPM Tripwire à partir des CD-ROM de Red Hat Linux 9.

Si vous n'êtes pas sûr que Tripwire est installé, tapez la commande suivante à l'invite du shell:

```
rpm -q tripwire
```

Si Tripwire est installé, cette commande fournira la réponse suivante:

```
tripwire-<version-number>
```

Dans la sortie ci-dessus, <*version-number*> correspond au numéro de la version du paquetage.

Si tripwire est installé, l'invite du shell réapparaîtra.

Les étapes suivantes définissent la manière de trouver et d'installer Tripwire à partir du CD-ROM en utilisant l'application de ligne de commande RPM:

1. Insérez le *CD-ROM 2* des CD-ROM d'installation de Red Hat Linux 9.
2. Si le CD-ROM n'est pas monté automatiquement, tapez la commande suivante:
   ```
   mount /mnt/cdrom
   ```
3. Vérifiez que le RPM de Tripwire figure sur le CD-ROM en tapant:
   ```
   ls /mnt/cdrom/RedHat/RPMS/ | grep tripwire
   ```
   Si le RPM est sur le CD-ROM, cette instruction affichera le nom du paquetage.

   Si le RPM *n'* est *pas* sur le CD-ROM, l'invite du shell réapparaîtra. Dans ce cas, vous devrez vérifier les autres CD-ROM d'installation de Red Hat Linux 9 en démontant d'abord le CD-ROM et en répétant ensuite les étapes un à trois.

   Démontez le CD-ROM en cliquant sur l'icône CD-ROM à l'aide du bouton droit de votre souris et en sélectionnant **Éjecter** ou en tapant la commande suivante:
   ```
   umount /mnt/cdrom
   ```
4. Après avoir localisé le RPM de Tripwire, installez-le en tapant la commande suivante en tant que root:
   ```
   rpm -Uvh /mnt/cdrom/RedHat/RPMS/tripwire*.rpm
   ```

Vous trouverez des notes et les fichiers README (Lisez-moi) concernant Tripwire dans le répertoire `/usr/share/doc/tripwire-<version-number>/` (sachant que `<version-number>` correspond au numéro de version du logiciel). Ces documents contiennent d'importantes informations concernant le fichier de politiques par défaut et d'autres sujets.

19.3. Personnalisation de Tripwire

Après avoir installé le RPM de Tripwire, vous devez suivre les étapes suivantes pour initialiser le logiciel:

19.3.1. Éditer `/etc/tripwire/twcfg.txt`

Bien que vous ne soyez pas obligé de modifier cet exemple de fichier de configuration Tripwire, cela peut s'avérer nécessaire dans votre situation. Par exemple, il est possible que vous vouliez modifier l'emplacement de fichiers Tripwire, personnaliser des paramètres d'email, ou personnaliser le niveau de détail des rapports.

Vous trouverez ci-dessous une liste des variables configurables par l'utilisateur *nécessaires* dans le fichier `/etc/tripwire/twcfg.txt`:

- `POLFILE` — Précise l'emplacement du fichier de politiques; `/etc/tripwire/tw.pol` est la valeur défaut.
- `DBFILE` — Précise l'emplacement du fichier de base données; `/var/lib/tripwire/$(HOSTNAME).twd` est la valeur par défaut.
- `REPORTFILE` — Précise l'emplacement du des fichiers rapport. Par défaut, cette valeur est réglée sur `/var/lib/tripwire/report/$(HOSTNAME)-$(DATE).twr`.
- `SITEKEYFILE` — Précise l'emplacement du fichier clé site; `/etc/tripwire/site.key` est la valeur par défaut.
- `LOCALKEYFILE` — Précise l'emplacement du fichier clé local; `/etc/tripwire/$(HOSTNAME)-local.key` est la valeur par défaut.



Important

> Si vous modifiez le fichier de configuration et laissez non-définie l'une de ces variables, le fichier de configuration sera considéré comme invalide. Si cela se produit lorsque vous exécutez la commande `tripwire` le fichier rapportera une erreur et se fermera.

Le reste des variables configurables dans l'exemple de fichier `/etc/tripwire/twcfg.txt` est optionnel. Ces variables comprennent:

- `EDITOR` — Précise l'éditeur de texte appelé par Tripwire. La valeur par défaut est `/bin/vi`.
- `LATEPROMPTING` — Si elle est réglée sur `true`, cette variable configure Tripwire pour attendre aussi longtemps que possible avant de demander à l'utilisateur un mot de passe, réduisant ainsi le temps pendant lequel le mot de passe est en mémoire. La valeur par défaut est `false`.
- `LOOSEDIRECTORYCHECKING` — Si elle est réglée sur `true`, cette variable configure Tripwire pour rapporter un fichier dans le cadre de modifications d'un répertoire observé et non pas pour rapporter la modification elle-même. Cela limite les répétitions dans les rapports Tripwire. La valeur par défaut est `false`.
- `SYSLOGREPORTING` — Si elle est réglée sur `true`, cette variable configure Tripwire pour rapporter des informations au démon système via l'option "utilisateur". Le niveau d'inscription est réglé sur `notice`. Consultez la page manuel `syslogd` pour obtenir davantage d'informations. La valeur par défaut est `false`.
- `MAILNOVIOLATIONS` — Si elle est réglée sur `true`, cette variable configure Tripwire pour envoyer un email de rapport à intervalles réguliers sans tenir compte des violations éventuellement survenues. La valeur par défaut est `true`.
- `EMAILREPORTLEVEL` — Précise le niveau de détail des rapports par email. Les valeurs valides pour cette variable vont de `0` à `4`. La valeur par défaut est `3`.
- `REPORTLEVEL` — Précise le niveau de détail des rapports générés par la commande `twprint`. Cette valeur peut être modifiée sur la ligne de commande, mais elle est réglée sur `3` par défaut.
- `MAILMETHOD` — Précise le protocole d'email que Tripwire devrait utiliser. Les valeurs acceptables sont `SMTP` et `SENDMAIL`. La valeur par défaut est `SENDMAIL`.
- `MAILPROGRAM` — Précise le programme d'email que Tripwire devrait utiliser. La valeur par défaut est `/usr/sbin/sendmail -oi -t`.

Après avoir modifié l'exemple de fichier de configuration, vous devrez configurer l'exemple de fichier politiques.



Avertissement

> Pour des raisons de sécurité, vous devriez soit supprimer, soit stocker dans un endroit sûr toute le copie du fichier texte `/etc/tripwire/twcfg.txt` après avoir exécuté le script d'installation ou avoir recréé un fichier de configuration signé. Vous pouvez également modifier les permissions de façon à ce qu'il soit illisible par toute personne non-autorisée.

19.3.2. Éditer `/etc/tripwire/twpol.txt`

Bien que cela ne soit pas nécessaire, vous pouvez modifier ce fichier politiques Tripwire comportant de nombreux commentaires pour prendre en considération les applications, les fichiers et les répertoires spécifiques sur votre système. Se fier à la configuration non-modifiée du RPM peut ne pas protéger votre système correctement.

Modifier le fichier politiques augmente également l'utilité des rapports de Tripwire en réduisant les fausses alertes pour les fichiers et programmes que vous n'utilisez pas et en ajoutant de la fonctionnalité, telle que notification par email.



Remarque

La notification via email n'est pas configurée par défaut. Consultez la Section 19.8.1 pour en savoir plus sur cette configuration.

Si vous modifiez l'exemple de fichier de politiques après avoir exécuté le script de configuration, veuillez lire la Section 19.8 pour savoir comment recréer un fichier de politiques signé.



Attention

Pour des raisons de sécurité, vous devez soit détruire soit stocker en lieu sûr toutes les copies texte du fichier `/etc/tripwire/twpol.txt` après l'exécution du script d'installation ou la création d'un fichier de configuration signé. Vous pouvez également changer les permissions de façon à le rendre illisible.

19.3.3. Exécution du script `twinstall.sh`

En tant qu'utilisateur root, tapez `/etc/tripwire/twinstall.sh` à l'invite du shell afin d'exécuter le script de configuration. Le script `twinstall.sh` vous demandera d'entrer votre mot de passe site et votre mot de passe local. Ces mots de passe sont utilisés pour créer des clés cryptographiques destinées à protéger les fichiers Tripwire. Le script crée alors ces fichiers puis les signe.

Lorsque vous choisissez les mots de passe site et local, vous devez respecter les indications suivantes:

- Utilisez au moins huit caractères alphanumériques et symboliques, mais ne dépassez pas 1023 pour chaque mot de passe.
- N'utilisez pas de citation dans les mots de passe.
- Les mots de passe Tripwire doivent être complètement différents du mot de passe root ou de tout autre mot de passe du système.
- Utilisez des mots de passe uniques pour la clé site et la clé locale.

Le mot de passe clé du site protège les fichiers de configuration et de politiques Tripwire. Le mot de passe clé local protège les fichiers de base de données et de rapports Tripwire.



Attention

Il n'existe aucun moyen de décrypter un fichier signé si vous oubliez votre mot de passe. Si vous oubliez les mots de passe, les fichiers sont inutilisables et vous devrez exécuter le script de configuration une nouvelle fois.

En cryptant ses fichiers de configuration, politiques, base de données et rapports, Tripwire les empêche d'être visualisés par quiconque ne connaît pas le site et les mots de passe locaux. Cela signifie que, même si un intrus obtiens l'accès root à votre système, il ne pourra pas modifier les fichiers Tripwire pour masquer la trace.

Une fois cryptés et signés, les fichiers configuration et politiques créés en exécutant le script `twinstall.sh`ne doivent être ni renommés ni déplacés.

19.4. Initialisation de la base de données de Tripwire

Lorsque la base de donnée est initialisée, Tripwire crée un ensemble d'objets du système de fichiers en se basant sur les règles contenues dans le fichier de politiques. Cette base de données est utilisée comme référence lors des vérifications d'intégrité.

Pour initialiser la base de données de Tripwire, utilisez la commande suivante:

```
/usr/sbin/tripwire --init
```

L'exécution de cette commande peut prendre un certain temps.

Lorsque vous avez exécuté ces étapes avec succès, Tripwire dispose d'un instantané référentiel de votre système de fichiers nécessaire pour vérifier les modifications de fichiers importants. Après initialisation de la base de données Tripwire, vous devriez exécuter une première vérification d'intégrité. Cette vérification doit être effectuée avant de relier l'ordinateur au réseau, et de le mettre en phase de production. Pour obtenir des instructions sur la manière de procéder, consultez la Section 19.5.

Une fois que Tripwire est configuré en fonction de vos besoins, vous pouvez commencer à utiliser le système.

19.5. Exécution d'une vérification d'intégrité

Par défaut, le Tripwire RPM ajoute un script de promptage appelé `tripwire-check` au répertoire `/etc/cron.daily/`. Ce script déclenchera automatiquement une vérification d'intégrité par jour.

Vous pouvez, toutefois, exécuter une vérification d'intégrité Tripwire à tout moment en tapant la commande suivante:

```
/usr/sbin/tripwire --check
```

Lors d'une vérification d'intégrité, Tripwire compare les objets actuels du système de fichiers avec leurs propriétés, qui sont enregistrées dans la base de données. Les violations sont imprimées à l'écran et une copie cryptée du rapport est créée dans `/var/lib/tripwire/report/`. Vous pouvez visualiser le rapport en utilisant la commande `twprint` comme nous l'avons décrit dans la Section 19.6.1.

Si vous souhaitez recevoir un email lorsque certains types de violations d'intégrité se produisent, vous pouvez le configurer dans le fichier politiques. Consultez la Section 19.8.1 pour obtenir des instructions sur la façon de régler et de tester cette option.

19.6. Examen des rapports Tripwire

La commande `/usr/sbin/twprint` est utilisée pour consulter les rapports et les bases de données cryptés de Tripzire .

19.6.1. Affichage des rapports de Tripwire

La commande `twprint -m r` affichera le contenu d'un rapport Tripwire en texte en clair. Vous devez toutefois préciser à `twprint` quel rapport doit être affiché.

Une commande `twprint` pour imprimer des rapports Tripwire ressemble à l'extrait ci-dessous:

```
/usr/sbin/twprint -m r --twrfile
/var/lib/tripwire/report/<name>.twr
```

L'option `-m r` de cette commande indique à `twprint` de décoder un rapport Tripwire. L'option `--twrfile` indique à `twprint` d'utiliser un fichier rapport Tripwire spécifique.

Le nom du rapport Tripwire que vous voulez visualiser contient le nom de l'hôte que Tripwire a contrôlé pour générer le rapport, ainsi que la date et l'heure de sa création. Vous pouvez à tout moment consulter des rapports enregistrés précédemment. Pour cela, vous n'avez qu'à taper `ls /var/lib/tripwire/report` pour faire apparaître une liste de rapports Tripwire.

Les rapports Tripwire peuvent être assez longs, selon le nombre de violations trouvées ou d'erreurs générées. Un exemple de rapport commence comme l'extrait ci-dessous:

```
Tripwire(R) 2.3.0 Integrity Check Report

Report generated by:          root
Report created on:            Fri Jan 12 04:04:42 2001
Database last updated on:     Tue Jan  9 16:19:34 2001

=======================================================================
Report Summary:
=======================================================================
Host name:                    some.host.com
Host IP address:              10.0.0.1
Host ID:                      None
Policy file used:             /etc/tripwire/tw.pol
Configuration file used:      /etc/tripwire/tw.cfg
Database file used:           /var/lib/tripwire/some.host.com.twd
Command line used:            /usr/sbin/tripwire --check

=======================================================================
Rule Summary:
=======================================================================
-----------------------------------------------------------------------
Section: Unix File System
-----------------------------------------------------------------------
  Rule Name                 Severity Level    Added    Removed  Modified
  ---------                 --------------    -----    -------  --------
  Invariant Directories     69                0        0        0
  Temporary directories     33                0        0        0
* Tripwire Data Files       100               1        0        0
  Critical devices          100               0        0        0
  User binaries             69                0        0        0
  Tripwire Binaries         100               0        0        0
```

19.6.2. Affichage des base de données de Tripwire

Vous pouvez également utiliser `twprint` pour visualiser la base de données complète ou certaines informations sur des fichiers de votre choix dans la base de données de Tripwire. C'est très pratique pour avoir une idée de la quantité d'informations contrôlées par Tripwire sur votre système.

Pour visualiser la base de données complète de Tripwire, entrez cette commande:

```
/usr/sbin/twprint -m d --print-dbfile | less
```

Cette commande créera une grande quantité de données et les premières lignes que vous verrez ressembleront à l'extrait ci-dessous:

```
Tripwire(R) 2.3.0 Database
```

```
Database generated by:        root
Database generated on:        Tue Jan  9 13:56:42 2001
Database last updated on:     Tue Jan  9 16:19:34 2001

=================================================================
Database Summary:
=================================================================
Host name:                    some.host.com
Host IP address:              10.0.0.1
Host ID:                      None
Policy file used:             /etc/tripwire/tw.pol
Configuration file used:      /etc/tripwire/tw.cfg
Database file used:           /var/lib/tripwire/some.host.com.twd
Command line used:            /usr/sbin/tripwire --init

=================================================================
Object Summary:
=================================================================
-----------------------------------------------------------------
# Section: Unix File System
-----------------------------------------------------------------
     Mode        UID          Size       Modify Time
     ------      ----------   ---------- ----------
 /
     drwxr-xr-x  root (0)     XXX        XXXXXXXXXXXXXXXXX
 /bin
     drwxr-xr-x  root (0)     4096       Mon Jan  8 08:20:45 2001
 /bin/arch
     -rwxr-xr-x  root (0)     2844       Tue Dec 12 05:51:35 2000
 /bin/ash
     -rwxr-xr-x  root (0)     64860      Thu Dec  7 22:35:05 2000
 /bin/ash.static
     -rwxr-xr-x  root (0)     405576     Thu Dec  7 22:35:05 2000
```

Pour avoir des renseignements sur un fichier en particulier, contrôlé par Tripwire, tel que `/etc/hosts`, tapez la commande suivante:

```
/usr/sbin/twprint -m d --print-dbfile /etc/hosts
```

Le résultat obtenu ressemblera à l'extrait ci-dessous:

```
Object name:  /etc/hosts

Property:                Value:
-------------            -----------
Object Type              Regular File
Device Number            773
Inode Number             216991
Mode                     -rw-r--r--
Num Links                1
UID                      root (0)
GID                      root (0)
```

Consultez la page de manuel relative à `twprint` pour obtenir des informations sur des options supplémentaires.

19.7. Mise à jour de la base de données de Tripwire

Si vous exécuter une vérification d'intégrité et que Tripwire décèle des violations du système vous devrez d'abord déterminer si ces violations sont causées par de véritables infractions au système de sécurité ou si elles sont provoquées de façon autorisée. Si, par exemple, vous avez récemment installé une application ou modifié des fichiers système critiques, Tripwire rapporte (avec raison) ces violations lors de la vérification d'intégrité. Dans ce cas précis, vous devez mettre à jour votre base de données Tripwire afin que ces changements ne soient plus considérés comme des violations du système. Toutefois, si des changements non-autorisés ont été apportés à des fichiers système et provoquent des violations lors de la vérification d'intégrité, devez alors restaurer les fichiers originaux à partir d'une copie de sauvegarde ou réinstaller le programme.

Pour mettre à jour votre base de données Tripwire, afin qu'elle accepte les violations de politiques valides, Tripwire compare d'abord un fichier de rapport avec la base de données, puis y intègre les violations valides depuis le fichier de rapport. Lorsque vous mettez à jour la base de données, assurez-vous d'utiliser le rapport le plus récent.

Utilisez la commande suivante pour mettre à jour la base de données Tripwire. Dans cette commande, `name` correspond au nom du fichier de rapport le plus récent:

```
/usr/sbin/tripwire --update --twrfile
/var/lib/tripwire/report/<name>.twr
```

Tripwire affichera le rapport au moyen de l'éditeur de texte par défaut spécifié dans le fichier de configuration de Tripwire à la ligne `EDITOR`. C'est à ce moment que vous avez la possibilité de désélectionner les fichiers que vous ne désirez pas inclure dans la mise à jour de la base de données Tripwire.

**Important**

Il est important de ne permettre que les modification des violations *autorisées* du système dans la base de données.

Toutes les mises à jour proposées de la base de données Tripwire commencent avec un `[x]` avant le nom du fichier, semblable à l'exemple suivant:

```
Added:
[x] "/usr/sbin/longrun"

Modified:
[x] "/usr/sbin"
[x] "/usr/sbin/cpqarrayd"
```

Si vous voulez spécifiquement exclure une violation valide afin qu'elle ne fasse pas partie de la mise à jour de la base de données Tripwire, enlevez le `x`.

Pour modifier des fichiers dans l'éditeur de texte par défaut, `vi`, tapez `i` puis appuyez sur la touche [Entrée] pour entrer en mode insertion et réaliser les changements nécessaires. Finalement, appuyez sur la touche [Échap], tapez `:wq` et appuyez sur [Entrée].

Après la fermeture de l'éditeur, entrez votre mot de passe local et la base données sera reconstruite et signée.

Une fois la nouvelle base de données Tripwire créée, les violations d'intégrité venant tout juste d'être autorisées ne seront plus indiquées comme des avertissements.

19.8. Mise à jour du fichier de politiques

Si vous désirez changer les fichiers que Tripwire enregistre dans sa base de données, changer la configuration de votre courrier électronique ou modifier la sévérité avec laquelle les violations sont rapportées, vous devez modifier le fichier de politiques de Tripwire.

Premièrement, apportez tous les changements nécessaires à l'exemple de fichier de politiques, /etc/tripwire/twpol.txt. Si vous avez effacé ce fichier (comme c'est le cas lorsque vous avez terminé de configurer Tripwire), vous pouvez le recréer en exécutant la commande suivante:

```
twadmin --print-polfile > /etc/tripwire/twpol.txt
```

L'un des changements couramment apportés à ce fichier est marquer tous les fichiers qui n'existent pas sur le système, afin qu'ils ne n'engendre pas un message d'erreur du type file not found dans les rapports de tripwire. Si, par exemple, votre système ne possède pas le fichier /etc/smb.conf, vous pouvez spécifier à Tripwire de ne pas essayer de le trouver en mettant le signe # sur sa ligne contenue dans le fichier twpol.txt, comme le montre l'exemple suivant:

```
#     /etc/smb.conf                        -> $(SEC_CONFIG) ;
```

Ensuite, vous devez indiquer à Tripwire de générer d'une part un nouveau fichier /etc/tripwire/tw.pol signé et d'autre part, une mise à jour du fichier de la base de données en fonction des nouvelles informations contenues dans le fichier de politiques. Si par exemple /etc/tripwire/twpol.txt était le fichier de politiques modifié, il faudrait utiliser la commande suivante:

```
/usr/sbin/twadmin --create-polfile -S site.key /etc/tripwire/twpol.txt
```

Vous devrez alors entrer le mot de passe donnant l'accès au site. Le fichier twpol.txt sera alors crypté et signé.

Il est important que vous mettiez à jour votre base de données Tripwire après la création d'un nouveau fichier /etc/tripwire/tw.pol. La façon la plus sûre de le faire consite à éliminer votre base de données Tripwire existante et d'en créer une nouvelle au moyen du nouveau fichier de politiques.

Si votre fichier de base de données Tripwire s'appelle bob.domain.com.twd, tapez la commande suivante:

```
rm /var/lib/tripwire/bob.domain.com.twd
```

Tapez ensuite la commande suivante pour créer une nouvelle base de données:

```
/usr/sbin/tripwire --init
```

Pour vous assurer que la base de données a été correctement modifiée, lancez une vérification d'intégrité manuelle et visualisez le contenu du rapport qui en résulte. Consultez la Section 19.5 et la Section 19.6.1 pour de plus amples informations sur la façon de procéder pour effectuer ces tâches.

19.8.1. Tripwire et Email

Vous pouvez configurer Tripwire pour envoyer un email à un pu plusieurs comptes si un type de fichier spécifique est violé. Pour ce faire, vous devez déterminer quelles règles de politiques doivent être contrôlées et qui doit être le destinataire du message lorsque ces règles sont violées. Notez également que sur les systèmes importants ayant plusieurs administrateurs système, vous pouvez faire en sorte que des groupes d'individus différents soient avertis selon le type de violations.

Une fois que vous savez qui avertir et quelles violations de règles doivent faire l'objet d'un rapport, modifiez le fichier /etc/tripwire/twpol.txt en ajoutant une ligne **emailto=** à la section de directive de règle correspondant à chaque règle concernée. Pour cela, ajoutez une virgule après la

ligne **severity=** et indiquez **emailto=** sur la ligne suivante, suivie par une ou plusieurs adresses email. Plusieurs adresses email peuvent être spécifiées à condition qu'elles soient séparées par un point-virgule.

Par exemple, si vous désirez avertir deux administrateurs, Sam et Bob, lorsqu'un programme de connexion au réseau est modifié, changez la directive de la règle des programmes de connexion au réseau dans le fichier de politiques afin qu'elle ressemble à l'extrait ci-dessous: :

```
(
  rulename = "Networking Programs",
  severity = $(SIG_HI),
  emailto = johnray@domain.com;bob@domain.com
)
```

Après modification du fichier de politiques, suivez les instructions contenue dans la Section 19.8 pour créer une copie mise à jour, cryptée et signée du fichier de politiques Tripwire.

19.8.1.1. Envoi de messages électroniques test

Afin de tester la configuration de la notification par email de Tripwire, utilisez la commande suivante:

```
/usr/sbin/tripwire --test --email
your@email.address
```

Un message test est ainsi envoyé immédiatement à l'adresse email par le programme `tripwire`.

19.9. Mise à jour du fichier de configuration Tripwire

Si vous désirez modifier le fichier de configuration de Tripwire, vous devez d'abord modifier l'exemple de fichier de configuration `/etc/tripwire/twcfg.txt`. Si vous avez détruit ce fichier (comme c'est le cas lorsque vous avez terminé de configurer Tripwire), vous pouvez le recréer en utilisant la commande suivante:

```
twadmin --print-cfgfile > /etc/tripwire/twcfg.txt
```

Tripwire ne reconnaîtra aucun changement de configuration tant que le fichier texte de configuration ne sera pas correctement signé et converti en `/etc/tripwire/tw.pol` avec la commande `twadmin`.

Utilisez la commande suivante pour recréer un fichier de configuration à partir du fichier texte `/etc/tripwire/twcfg.txt`:

```
/usr/sbin/twadmin --create-cfgfile -S site.key /etc/tripwire/twcfg.txt
```

Étant donné que le fichier de configuration ne modifie pas les politiques Tripwire ou les fichiers contrôlés par l'application, il n'est pas nécessaire de recréer la base données.

19.10. Référence d'emplacement de fichier Tripwire

Avant de travailler avec Tripwire, vous devez savoir où sont situés les fichiers importants pour l'application. Tripwire stocke ses fichiers dans un grand nombre d'endroits qui dépendent de leur rôle.

- Dans le répertoire `/usr/sbin/` vous trouverez les programmes suivants:
  - `tripwire`

- `twadmin`
- `twprint`

- Dans le répertoire `/etc/tripwire/` vous trouverez les fichiers suivants:
  - `twinstall.sh` — Le script d'initialisation pour Tripwire.
  - `twcfg.txt` — L'exemple de fichier de configuration fourni par le RPM de Tripwire.
  - `tw.cfg` — Le fichier de configuration signé créé par le `twinstall.sh` script.
  - `twpol.txt` — L'exemple de fichier de politiques fourni par le RPM de Tripwire.
  - `tw.pol` — Le fichier de politiques signé créé par le script `twinstall.sh`.
  - Fichiers de clés — Les clés locale et site créées par le script `twinstall.sh` qui fini par l'extension de fichier `.key`.

- Après exécution du script d'installation `twinstall.sh`, vous trouverez les fichiers suivants dans le répertoire `/var/lib/tripwire/`:
  - La base de données Tripwire — La base de données des fichiers de votre système qui a une extension de fichier `.twd`.
  - Rapports Tripwire — Le répertoire `report/` est l'endroit où sont stockés les rapports Tripwire.

La section suivante donne davantage de détails au sujet des rôles joués par ces fichiers dans le système Tripwire.

19.10.1. Composants de Tripwire

Ce qui suit décrit de façon plus détaillée les rôles - dont la liste figure dans la section précédente - joués par les fichiers dans le système Tripwire.

/etc/tripwire/tw.cfg

Il s'agit du fichier de configuration Tripwire crypté qui stocke les informations spécifiques au système, telles que l'emplacement des fichiers de données Tripwire. Le script d'installation `twinstall.sh` et la commande `twadmin` crée ce fichier en utilisant les informations de la version texte du fichier de configuration, `/etc/tripwire/twcfg.txt`.

Après avoir exécuté le script d'installation, l'administrateur système peut changer les paramètres en modifiant `/etc/tripwire/twcfg.txt` et en recréant une copie signée du fichier `tw.cfg` en utilisant la commande `twadmin`. Pour obtenir davantage d'informations sur la façon de procéder, reportez-vous à la Section 19.9.

/etc/tripwire/tw.pol

Le fichier de politiques actif Tripwire est un fichier crypté contenant des commentaires, des règles, des directives et des variables. Ce fichier commande la façon dont Tripwire vérifie votre système. Chaque règle du fichier politiques spécifie un objet système à vérifier. Les règles décrivent également les changements d'objet à rapporter et ceux qui sont à ignorer.

Les objets système sont les fichiers et les répertoires que vous souhaitez contrôler. Chaque objet est identifié par un nom d'objet. Une propriété se réfère à une caractéristique unique d'un objet que le logiciel Tripwire peut contrôler. Les directives contrôlent le processus conditionnel des séries de règles d'un fichier politiques. Au cours de l'installation, l'exemple de fichier politiques texte `/etc/tripwire/twpol.txt`, est utilisé pour créer un fichier politiques Tripwire actif.

Après l'exécution du script d'installation, l'administrateur système peut mettre à jour le fichier politiques Tripwire en modifiant `/etc/tripwire/twpol.txt` et en recréant une copie signée du fichier `tw.pol` à l'aide de la commande `twadmin`. Pour obtenir davantage d'informations sur la façon de procéder, reportez-vous à la Section 19.8.

/var/lib/tripwire/host_name.twd

Lorsqu'il est initialisé pour la première fois, Tripwire utilise les règles du fichier de politiques signé pour créer ce fichier de la base de données. La base de données Tripwire est un instantané de référence du système à un état sûr connu. Tripwire compare ce fichier référentiel avec le système en cours pour déterminer si des changements ont eu lieu. Cette comparaison est appelée *vérification d'intégrité*.

/var/lib/tripwire/report/host_name-date_of_report-time_of_report.twr

Lorsque vous effectuez une vérification d'intégrité, Tripwire produit des fichiers rapport, placés dans le répertoire `/var/lib/tripwire/report/`. Ces fichiers rapport indiquent toutes les modifications apportées aux fichiers violant les règles du fichier de politiques lors de la vérification d'intégrité. Le nom des rapports de Tripwire est attribué en utilisant la convention suivante: *host_name-date_of_report-time_of_report*`.twr`. Ces rapports détaillent les différences entre la base données Tripwire et vos fichiers systèmes.

19.11. Ressources supplémentaires

Les capacités de Tripwire vont au-delà de ce qui est abordé dans ce chapitre. Reportez-vous aux ressources supplémentaires pour obtenir davantage d'informations sur Tripwire.

19.11.1. Documentation installée

- `/usr/share/doc/tripwire-`<*version-number*> — Un excellent point de départ pour apprendre à personnaliser les fichiers de configuration et de politiques dans le répertoire `/etc/tripwire/`.
- Pour obtenir de l'aide sur l'utilisation de ces programmes utilitaires, lisez aussi les pages de manuel relatives à `tripwire`, `twadmin` et `twprint`.

19.11.2. Sites Web utiles

- http://www.tripwire.org — Le site Web du projet Open Source Tripwire, où vous trouverez les toutes dernières nouvelles sur cette application et un Forum aux Questions très utile.
- http://sourceforge.net/project/showfiles.php?group_id=3130 — Ce lien renvoie à la dernière documentation officielle au sujet du projet Tripwire.

IV. Annexes

Table des matières



Annexe A. Paramètres généraux et modules

Cette annexe est fournie pour illustrer *certains* des paramètres dont des *pilotes*[1] de périphériques matériels courants peuvent avoir besoin et qui sous Red Hat Linux sont appelés *modules* du noyau. Dans la plupart des cas, les paramètres par défaut permettrons un bon fonctionnement. Néanmoins, dans certaines situations, des paramètres de module supplémentaires sont nécessaires afin qu'un périphérique puisse fonctionner correctement, ou s'il est nécessaire d'outrepasser les paramètres par défaut du module pour ce périphérique.

Durant l'installation, Red Hat Linux utilise un jeu limité de pilotes de périphériques afin de créer un environnement d'installation stable. Bien que le programme d'installation permette une installation sur des types de matériel très variés, certains pilotes (y compris ceux avec pour des adaptateurs SCSI, réseau et de nombreux lecteurs de CD-ROM) ne sont pas inclus dans le noyau d'installation. Ils doivent donc être chargés par l'utilisateur comme des modules lors du démarrage. Pour des informations sur l'endroit où trouver les modules du noyau supplémentaires lors du processus d'installations, reportez-vous à la section relative aux méthodes de démarrages supplémentaires présente dans le chapitre intitulé *Étapes pour démarrer* du *Guide d'installation de Red Hat Linux*.

Une fois l'installation terminée, la prise en charge de nombreux périphériques est assurée grâce à des modules du noyau.

A.1. Spécification des paramètres d'un module

Dans certaines situations, il peut s'avérer nécessaire de fournir certains paramètres à un module lors de son chargement, afin qu'il puisse fonctionner correctement. Ces informations peuvent être fournies de deux manières:

- Il est possible de spécifier un ensemble complet de paramètres au moyen d'une seule instruction. Par exemple, le paramètre `cdu31=0x340,0` pourrait être utilisé avec un CDU Sony 31 ou 33 sur le port 340 sans IRQ.
- Il est également possible de spécifier les paramètres de façon individuelle. Cette méthode est utilisée lorsqu'un ou plusieurs paramètres du premier ensemble ne sont pas nécessaires. Par exemple, `cdu31_port=0x340 cdu31a_irq=0` peut être utilisé comme paramètre pour le même lecteur de CD-ROM. On utilise un *OR* dans les tableaux CD-ROM, SCSI et Ethernet de cette annexe pour montrer où la première méthode de paramétrage s'arrête et où la seconde commence.



Remarque

N'utilisez qu'une seule méthode, et non pas les deux, lorsque vous chargez un module avec des paramètres spécifiques.

1. Un pilote est un type de logiciel qui permet au système Linux d'utiliser un périphérique matériel donné. Sans ce pilote, le noyau ne peut pas communiquer avec les périphérique qui lui sont attachés.



Avertissement

Lorsqu'un paramètre contient une virgule, assurez-vous de *ne pas* mettre d'espace après la virgule.

A.2. Paramètres des modules pour CD-ROM



Remarque

Les lecteurs de CD-ROM répertoriés ne sont pas tous pris en charge. Veuillez consulter la liste de compatibilité des composants matériels sur le site Web de Red Hat à l'adresse suivante: http://hardware.redhat.com pour vous assurer que votre lecteur de CD-ROM est bien pris en charge.

Bien que les paramètres soient spécifiés une fois la disquette de pilotes chargée et le périphérique défini, l'un des paramètres les plus couramment utilisés, `hdX=cdrom` (où *X* correspond à la lettre identifiant le périphérique approprié) *peut* être entré à une invite de démarrage (boot:) lors de l'installation. Ceci est permis car il a une influence sur les CD-ROM IDE/ATAPI CD-ROM, qui font déjà partie du noyau.

Dans les tableaux suivants, la plupart des modules répertoriés sans aucun paramètre sont capables d'une détection automatique du matériel ou, nécessiteront un changement manuel de paramètres dans le code source du module et une recompilation.

Matériel	Module	Paramètres
Lecteurs de CD-ROM ATAPI/IDE		hd*X*=cdrom
Aztech CD268-01A, Orchid CD-3110, Okano/Wearnes CDD110, Conrad TXC, CyCDROM CR520, CyCDROM CR540 (non IDE)	`aztcd.o`	aztcd=*io_port*
Sony CDU-31A CD-ROM	`cdu31a.o`	cdu31a=*io_port*,*IRQ* ***OU*** cdu31a_port=*base_addr* cdu31a_irq=*irq*
Lecteur de CD-ROM Philips/LMS 206 avec carte adaptateur hôte cm260	`cm206.o`	cm206=*io_port, IRQ*
Goldstar R420 CD-ROM	`gscd.o`	gscd=*io_port*
Interface CD-ROM de carte son ISP16, MAD16 ou Mozart (OPTi 82C928 et OPTi 82C929) avec lecteurs Sanyo/Panasonic, Sony ou Mitsumi	`isp16.o`	isp16=*io_port*,*IRQ*,*dma*, *drive_type* ***OR*** isp16_cdrom_base=*io_port* isp16_cdrom_irq=*IRQ* isp16_cdrom_dma=*dma* isp16_cdrom_type=*drive_type*
CD-ROM Mitsumi standard	`mcd.o`	mcd=*io_port*,*IRQ*

Matériel	Module	Paramètres
CD-ROM Mitsumi expérimental	`mcdx.o`	mcdx=*io_port_1*,*IRQ_1*, *io_port_n*,*IRQ_n*
Lecteur de CD-ROM de stockage optique "Dolphin" 8000 AT, Lasermate CR328A	`optcd.o`	
CD-ROM IDE port parallèle	`pcd.o`	
SB Pro 16 compatible	`sbpcd.o`	sbpcd=*io_port*
Sanyo CDR-H94A	`sjcd.o`	sjcd=*io_port* *OU* sjcd_base=*io_port*
CDU-535 et 531 de Sony (certains lecteurs Procomm)	`sonycd535.o`	sonycd535=*io_port*

Tableau A-1. Paramètres du matériel

Ci-après figurent des exemples de certains modules utilisés:

Configuration	Exemple
CD-ROM ATAPI, branché comme maître sur le deuxième canal IDE	hdc=cdrom
CD-ROM Mitsumi non-IDE sur port 340, IRQ 11	mcd=0x340,11
Trois lecteurs de CD-ROM Mitsumi non-IDE utilisant le pilote expérimental, les ports E/S 300, 304 et 320 avec IRQ 5, 10 et 11	mcdx=0x300,5,0x304,10,0x320,11
Sony CDU 31 ou 33 au port 340, sans IRQ	cdu31=0x340,0 *OU* cdu31_port=0x340 cdu31a_irq=0
CD-ROM Aztech sur port 220	aztcd=0x220
CD-ROM de type Panasonic sur interface SoundBlaster connecté au port 230	sbpcd=0x230,1
Phillips/LMS cm206 et cm260 à E/S 340 et IRQ 11	cm206=0x340,11
Goldstar R420 à IO 300	gscd=0x300
Lecteur Mitsumi sur carte son MAD16 à adresse ES 330 et IRQ 1, test DMA	isp16=0x330,11,0,Mitsumi
Sony CDU 531 à IO adresse 320	sonycd535=0x320

Tableau A-2. Exemples de configuration de paramètres matériels



Remarque

La plupart des cartes Sound Blaster récentes sont livrées avec des interfaces IDE. Pour ces cartes, vous ne devez pas utiliser de paramètres `sbpcd`; utilisez uniquement des paramètres `hdX` (où *X* correspond à la lettre identifiant le périphérique appropriés).

A.3. Paramètres SCSI

Matériel	Module	Paramètres
Adaptec 28xx, R9xx, 39xx	`aic7xxx.o`	
Contrôleur de mémoire 3ware	`3w-xxxx.o`	
NCR53c810/820/720, NCR53c700/710/700-66	`53c7,8xx.o`	
AM53/79C974 (PC-SCSI) Driver	`AM53C974.o`	
La plupart des cartes Buslogic (maintenant Mylex) avec numéro de référence "BT"	`BusLogic.o`	
Contrôleur Mylex DAC960 RAID	`DAC960.o`	
SCSI base sur MCR53c406a	`NCR53c406a.o`	
Initio INI-A100U2W	`a100u2w.o`	a100u2w=*io*,*IRQ*,*scsi_id*
AACRAID Adaptec	`aacraid.o`	
Cartes SCSI Advansys	`advansys.o`	
Adaptec AHA-152x	`aha152x.o`	aha152x=*io*,*IRQ*,*scsi_id*
AHA 154x et 631x de type Adaptec	`aha1542.o`	
Adaptec AHA 1740	`aha1740.o`	
Adaptec AHA-274x, AHA-284x, AHA-29xx, AHA-394x, AHA-398x, AHA-274x, AHA-274xT, AHA-2842, AHA-2910B, AHA-2920C, AHA-2930/U/U2, AHA-2940/W/U/UW/AU/, U2W/U2/U2B/, U2BOEM, AHA-2944D/WD/UD/UWD, AHA-2950U2/W/B, AHA-3940/U/W/UW/, AUW/U2W/U2B, AHA-3950U2D, AHA-3985/U/W/UW, AIC-777x, AIC-785x, AIC-786x, AIC-787x, AIC-788x , AIC-789x, AIC-3860	`aic7xxx.o`	
Contrôleur SCSI PCI ACARD ATP870U	`atp870u.o`	

Matériel	Module	Paramètres
Contrôleur Compaq Smart Array 5300	`cciss.o`	
Contrôleur Compaq Smart/2 RAID	`cpqarray.o`	
Contrôleur Compaq FibreChannel	`cpqfc.o`	
Domex DMX3191D	`dmx3191d.o`	
Data Technology Corp DTC3180/3280	`dtc.o`	
Cartes hôtes SCSI DTP (EATA/DMA)PM2011B/9X ISA, PM2021A/9X ISA, PM2012A, PM2012B, PM2022A/9X EISA, PM2122A/9X, PM2322A/9X, SmartRAID PM3021, PM3222, PM3224	`eata.o`	
Cartes SCSI DTP PM2011, PM2021, PM2041, PM3021, PM2012B, PM2022, PM2122, PM2322, PM2042, PM3122, PM3222, PM3332, PM2024, PM2124, PM2044, PM2144, PM3224, PM3334	`eata_dma.o`	
Sun Enterprise Network Array (FC-AL)	`fcal.o`	
Future Domain TMC-16*xx* SCSI	`fdomain.o`	
NCR5380 (pilote générique)	`g_NCR5380.o`	
Contrôleur ICP RAID	`gdth.o`	
I2O Block Driver (pilote de bloc)	`i2o_block.o`	
Carte SCSI pour port parallèle IOMEGA MatchMaker	`imm.o`	
Carte SCSI ISA Always IN2000	`in2000.o`	in2000=*setup_string*:*valeur* *OU* in2000 *setup_string*=*valeur*
Cartes hôtes SCSI Initio INI-9X00U/UW	`initio.o`	
IBM ServeRAID	`ips.o`	
AMI MegaRAID 418, 428, 438, 466, 762	`megaraid.o`	
Cartes SCSI NCR avec circuits 810/810A/815/825/825A/860/875/876/895	`ncr53c8xx.o`	ncr53c8xx=*option1*:*valeur1*, *option2*:*valeur2*,*...* *OU* ncr53c8xx="*option1*:*valeur1* *option2*:*valeur2*...*"*

Matériel	Module	Paramètres
Pro Audio Spectrum/Studio 16	`pas16.o`	
PCI-2000 IntelliCache	`pci2000.o`	
PCI-2220I EIDE RAID	`pci2220i.o`	
Carte hôte SCSI pour port parallèle IOMEGA PPA3	`ppa.o`	
Perceptive Solutions PSI-240I EIDE	`psi240i.o`	
Qlogic 1280	`qla1280.o`	
Qlogic 2*x*00	`qla2x00.o`	
QLogic Fast SCSI FASXXX ISA/VLB/PCMCIA	`qlogicfas.o`	
QLogic ISP2100 SCSI-FCP	`qlogicfc.o`	
Cartes SCSI QLogic ISP1020 Intelligent IQ-PCI, IQ-PCI-10, IQ-PCI-D	`qlogicisp.o`	
SBUS SCSI Qlogic ISP1020	`qlogicpti.o`	
Future Domain TMC-885, TMC-950 Seagate ST-01/02, Future Domain TMC-8*xx*	`seagate.o`	controller_type=2 base_address=*base_addr* irq=*IRQ*
Cartes avec circuit sym53c416	`sym53c416.o`	sym53c416=*PORTBASE*, *[IRQ]* *OU* sym53c416 io=*PORTBASE* irq=*IRQ*
Carte hôte SCSI Trantor T128/T128F/T228	`t128.o`	
Tekram DC-390(T) PCI	`tmscsim.o`	
UltraStor 14F/34F (not 24F)	`u14-34f.o`	
UltraStor 14F, 24F, and 34F	`ultrastor.o`	
Série WD7000	`wd7000.o`	

Tableau A-3. Paramètres SCSI

Ci-après figurent des exemples de certains modules utilisés:

Configuration	Exemple
Adaptec AHA1522 sur port 330, IRQ 11, SCSI ID 7	aha152x=0x330,11,7
Adaptec AHA1542 sur port 330	bases=0x330
Future Domain TMC-800 à CA000, IRQ 10	controller_type=2 base_address=0xca000 irq=10

Tableau A-4. Exemples de configuration des paramètres SCSI

A.4. Paramètres Ethernet



Important

De nos jours, la pluspart des cartes d'interface réseau basées sur Ethernet (NIC) ne nécessitent pas de paramètres de module pour modifier la configuration. Ils peuvent être configurés à l'aide de `ethtool` ou de `mii-tool`. Les paramètres de module ne devraient être ajustés que si ces outils ne fonctionnent pas.

Pour de plus amples informations sur l'utilisation de ces outils, reportez-vous aux pages de manuel relatives à `ethtool` et `mii-tool`.

Matériel	Module	Paramètres
3Com 3c501	`3c501.o`	3c501=*io_port*, *IRQ*
3Com 3c503 et 3c503/16	`3c503.o`	3c503=*io_port*,*IRQ* ***OR*** 3c503 io=*io_port_1*,*io_port_n* irq=*IRQ_1*, *IRQ_n*
3Com EtherLink Plus (3c505)	`3c505.o`	3c505=*io_port*, *IRQ* ***OR*** 3c505 io=*io_port_1*,*io_port_n* irq=*IRQ_1*,*IRQ_2*
3Com EtherLink 16	`3c507.o`	3c507=*io_port*, *IRQ* ***OU*** 3c507 io=*io_port* irq=*IRQ*
3Com EtherLink III	`3c509.o`	3c509=*io_port*, *IRQ*
3Com ISA EtherLink XL "Corkscrew"	`3c515.o`	
3Com EtherLink PCI III/XL Vortex (3c590, 3c592, 3c595, 3c597) Boomerang (3c900, 3c905, 3c595)	`3c59x.o`	full_duplex= *0* est actif *1* est inactif
RTL8139, SMC EZ Card Fast Ethernet	`8139too.o`	
Cartes RealTek utilisant RTL8129 ou circuits RTL8139 Fast Ethernet	`8139too.o`	
Apricot 82596	`82596.o`	
Ansel Communications Modèle 3200	`ac3200.o`	ac3200=*io_port*,*IRQ* ***OU*** ac3200 io=*io_port_1*,*io_port_n* irq=*IRQ_1*,*IRQ_n*
Alteon AceNIC Gigabit	`acenic.o`	
Aironet Arlan 655	`arlan.o`	
Allied Telesis AT1700	`at1700.o`	at1700=*io_port*,*IRQ* ***OU*** at1700 io=*io_port* irq=*IRQ*

Matériel	Module	Paramètres
Adaptateur Ethernet Broadcom BCM5700 10/100/1000	`bcm5700.o`	
Crystal SemiconductorCS89[02]0	`cs89x0.o`	
Cartes EtherWORKS DE425 TP/COAX EISA, DE434 TP PCI, DE435/450 TP/COAX/AUI PCI DE500 10/100 PCI Kingston, LinkSys, SMC8432, SMC9332, Znyx31[45], and Znyx346 10/100 cards with DC21040 (no SROM), DC21041[A], DC21140[A], DC21142, DC21143	`de4x5.o`	de4x5=*io_port* ***OU*** de4x5 io=*io_port* de4x5 args='eth*X[fdx]* autosense=*MEDIA_STRING*'
D-Link DE-600 Ethernet Pocket Adapter (Adaptateur de poche)	`de600.o`	
D-Link DE-620 Ethernet Pocket Adapter (Adaptateur de poche)	`de620.o`	
DIGITAL DEPCA & EtherWORKS DEPCA, DE100, DE101, DE200 Turbo, DE201Turbo DE202 Turbo TP/BNC, DE210, DE422 EISA	`depca.o`	depca=*io_port*,*IRQ* ***OU*** depca io=*io_port* irq=*IRQ*
Digi Intl. RightSwitch SE-X EISA et PCI	`dgrs.o`	
Davicom DM9102(A)/DM9132/ DM9801 Fast Ethernet	`dmfe.o`	
pILOTE Intel Ether Express/100	`e100.o`	e100_speed_duplex=*X* *If X* = 0 = autodetect speed and duplex 1 = 10Mbps, half duplex 2 = 10Mbps, full duplex 3 = 100Mbps, half duplex 4 = 100Mbps, full duplex
Intel EtherExpress/1000 Gigabit	`e1000.o`	
Cabletron E2100	`e2100.o`	e2100=*io_port,IRQ,mem* ***OU*** e2100 io=*io_port* irq=*IRQ* mem=*mem*
Intel EtherExpress Pro10	`eepro.o`	eepro=*io_port*,*IRQ* ***OU*** eepro io=*io_port* irq=*IRQ*

Matériel	Module	Paramètres
Pilote Intel i82557/i82558 PCI EtherExpressPro	`eepro100.o`	
Intel EtherExpress 16 (i82586)	`eexpress.o`	eexpress=*io_port*,*IRQ* ***OR*** eexpress io=*io_port* irq=*IRQ* options= 0x10 10base T half duplex 0x20 10base T full duplex 0x100 100base T half duplex 0x200 100baseT full duplex
SMC EtherPower II 9432 PCI (Série EPIC 83c170/175)	`epic100.o`	
Racal-Interlan ES3210 EISA	`es3210.o`	
ICL EtherTeam 16i/32 EISA	`eth16i.o`	eth16i=*io_port*,*IRQ* ***OU*** eth16i ioaddr=*io_port* IRQ=*IRQ*
EtherWORKS 3 (DE203, DE204 and DE205)	`ewrk3.o`	ewrk=*io_port*,*IRQ* ***OU*** ewrk io=*io_port* irq=*IRQ*
A Packet Engines GNIC-II Gigabit	`hamachi.o`	
HP PCLAN/plus	`hp-plus.o`	hp-plus=*io_port*,*IRQ* ***OR*** hp-plus io=*io_port* irq=*IRQ*
HP LAN Ethernet	`hp.o`	hp=*io_port*,*IRQ* ***OU*** hp io=*io_port* irq=*IRQ*
Cartes réseau 100VG-AnyLan HP J2585B, J2585A, J2970, J2973, J2573 Compex ReadyLink ENET100-VG4, FreedomLine 100/VG	`hp100.o`	hp100=*io_port*,*nom* ***OU*** hp100 hp100_port=*io_port* hp100_name=*nom*
IBM Token Ring 16/4, Shared-Memory IBM Token Ring 16/4 (mémoire partagée)	`ibmtr.o`	ibmtr=*io_port* ***OU*** io=*io_port*
AT1500, HP J2405A, et la plupart des clones NE2100	`lance.o`	
Mylex LNE390 EISA	`lne390.o`	
NatSemi DP83815 Fast Ethernet	`natsemi.o`	
NE1000 / NE2000 (non-pci)	`ne.o`	ne=*io_port*,*IRQ* ***OU*** ne io=*io_port* irq=*IRQ*

Matériel	Module	Paramètres
Cartes PCI NE2000 RealTEk RTL-8029, Winbond 89C940, Compex RL2000, PCI NE2000 clones, NetVin, NV5000SC, Via 82C926, SureCom NE34	`ne2k-pci.o`	
Novell NE3210 EISA	`ne3210.o`	
MiCom-Interlan NI5010	`ni5010.o`	
Carte NI5210 (puce Ethernet i82586)	`ni52.o`	ni52=*io_port*,*IRQ* ***OU*** ni52 io=*io_port* irq=*IRQ*
NI6510 Ethernet	`ni65.o`	
IBM Olympic-based PCI token ring	`olympic.o`	
AMD PCnet32 et AMD PCnetPCI	`pcnet32.o`	
SIS 900/701G PCI Fast Ethernet	`sis900.o`	
SysKonnect SK-98*XX* Gigabit	`sk98lin.o`	
SMC Ultra et SMC EtherEZ ISA ethercard (8K, 83c790)	`smc-ultra.o`	smc-ultra=*io_port*,*IRQ* ***OU*** smc-ultra io=*io_port* irq=*IRQ*
Carte Ethernet EISA SMC Ultra32 (32K)	`smc-ultra32.o`	
Sun BigMac Ethernet	`sunbmac.o`	
Sundance ST201 Alta	`sundance.o`	
Sun Happy Meal Ethernet	`sunhme.o`	
Sun Quad Ethernet	`sunqe.o`	
ThunderLAN	`tlan.o`	
Cartes Ethernet PCI Digital 21x4x Tulip SMC EtherPower 10 PCI(8432T/8432BT) SMC EtherPower 10/100 PCI(9332DST) DEC EtherWorks 100/10 PCI(DE500-XA) DEC EtherWorks 10 PCI(DE450) DEC QSILVER's, Znyx 312 etherarray Allied Telesis LA100PCI-T Danpex EN-9400, Cogent EM110	`tulip.o`	io=*io_port*

Matériel	Module	Paramètres
Cartes PCI Fast Ethernet VIA Rhine PCI avec soit VIA VT86c100A Rhine-II PCI ou 3043 Rhine-I D-Link DFE-930-TX PCI 10/100	`via-rhine.o`	
AT&T GIS (nee NCR) WaveLan ISA Card	`wavelan.o`	wavelan=*[IRQ,0],io_port, NWID*
WD8003 et Cartes compatibles Ethernet WD8013	`wd.o`	wd=*io_port*,*IRQ*,*mem*, *mem_end* *OU* wd io=*io_port* irq=*IRQ* mem=*mem* mem_end=*end*
Compex RL100ATX-PCI	`winbond.o`	
Packet Engines Yellowfin	`yellowfin.o`	

Tableau A-5. Paramètres de modules Ethernet

Ci-après figurent des exemples de certains modules utilisés:

Configuration	Exemple
Carte ISA NE2000 à l'adresse E/S 300 et IRQ 11	ne=0x300,11 ether=0x300,11,eth0
Carte Wavelan à l'E/S 390, détection automatique d'IRQ et utilisation de NWID pour 0x4321	wavelan=0,0x390,0x4321 ether=0,0x390,0x4321,eth0

Tableau A-6. Exemples de configuration de paramètres Ethernet

A.4.1. Utilisation de plusieurs cartes Ethernet

Vous pouvez utiliser plusieurs cartes Ethernet dans un ordinateur. Si chaque carte utilise un pilote différent (par exemple, 3c509 et DE425), vous devez simplement ajouter des lignes `alias` (et éventuellement `options`) pour chaque carte dans le fichier `/etc/modules.conf`. Reportez-vous au chapitre intitulé *Modules du noayu* du *Guide de personnalisation de Red Hat Linux* pour obtenir de plus amples informations à ce sujet.

Si deux cartes Ethernet utilisent le même pilote (par exemple, deux cartes 3c509 ou une 3c595 et une 3c905), vous devez soit indiquer les adresses des deux cartes dans la ligne d'options du pilote (pour les cartes ISA), soit simplement ajouter une ligne `alias` pour chaque carte (pour les cartes PCI).

Pour plus d'informations sur l'utilisation de plusieurs cartes Ethernet, consultez la section *Linux Ethernet-HOWTO* à l'adresse suivante: http://www.redhat.com/mirrors/LDP/HOWTO/Ethernet-HOWTO.html.

Index

Symbols

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

S

T

U

V

W

X



Colophon

Les guides Red Hat Linux sont écrits sous format DocBook SGML v4. Les formats HTML et PDF sont produits à l'aide de feuilles de style DSSSL personnalisées et de scripts de wrapper jade personnalisés. Les fichiers DocBook SGML sont écrits avec **Emacs** avec l'aide du mode PSGML.

Garrett LeSage a créé les graphiques d'admonition (remarque, astuce, important, attention et avertissement). Ils peuvent être librement redistribués avec la documentation Red Hat.

L'équipe de documentation de produits Red Hat Linux est composée des personnes suivantes:

Sandra A. Moore — Rédaction/Conception du *Guide d'installation de x86 Red Hat Linux*; Contribution à la rédaction du *Guide de démarrage de Red Hat Linux*

Tammy Fox — Rédaction/Conception du *Guide de personnalisation de Red Hat Linux*; Contribution à la rédaction du *Guide de démarrage de Red Hat Linux*; Rédaction/Conception des feuilles de style et des scripts DocBook personnalisés

Edward C. Bailey — Rédaction/Conception du *Guide d'administration système de Red Hat Linux*; Contribution à la rédaction du *Guide d'installation de x86 Red Hat Linux*

Johnray Fuller — Rédaction/Conception du *Guide de référence de Red Hat Linux*; Co-rédaction/Co-conception du *Guide de sécurité de Red Hat Linux*; Contribution à la rédaction du *Guide d'administration système de Red Hat Linux*

John Ha — Rédaction/Conception du *Guide de démarrage de Red Hat Linux*; Co-rédaction/Co-conception du *Guide de sécurité de Red Hat Linux*; Contribution à la rédaction du *Guide d'administration système de Red Hat Linux*

Jean-Paul Aubry — Traduction du *Guide d'installation de x86 Red Hat Linux*. Traduction du *Guide de démarrage de Red Hat Linux*. Traduction du *Guide de personnalisation de Red Hat Linux*. Traduction du *Guide de référence de Red Hat Linux*.